Rekursiivisten ja lambda-määriteltävien funktioiden yhtenevyys

Juho Kyyhkynen

Pro gradu -tutkielma

Helsingin Yliopisto
Matemaattis-luonnontieteellinen tiedekunta
Matematiikan ja tilastotieteen osasto
Marraskuu 2018

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Matemaattis-luonnontieteellinen	Matematiikan ja tilastotieteen osasto

Tekijä — Författare — Author

Juho Kyyhkynen

Työn nimi — Arbetets titel — Title

Rekursiivisten ja lambda-määriteltävien funktioiden yhtenevyys

Oppiaine — Läroämne — Subject

Matematiikka

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Pro gradu -tutkielma	Marraskuu 2018	41 s.

Tiivistelmä — Referat — Abstract

Tämän tutkielman päämäärä on osoittaa rekursiivisten funktioiden ja lambda-määriteltävien funktioiden yhtenevyys. Molemmat funktiojoukot ovat laskettavuuden teorian syntyyn vaikuttaneita laskennan malleja, joilla kuvataan automatisoitavissa olevia prosesseja.

Kurt Gödel tarvitsi rekursiivisia funktioita predikaattilogiikan todistuvuuden mekanisointiin. Lambda-määriteltävyys taas pohjautuu Alonzo Churchin ideoimaan lambdalaskentaan (engl. *lambda-calculus*). Se koostuu funktioita esittävistä kaavoista, ja niille määritellyistä muunnossäännöistä. Lambdalaskenta on toisinaan suomennettu myös lambdakalkyyliksi.

Lambda-määriteltävät funktiot on rekursiivisten funktioiden ohella osoitettu yhteneviksi useiden muidenkin laskentaformalismien kanssa. Tämän johdosta on päädytty otaksumaan, että rekursiiviset operaatiot ovat ne, joita on ylipäänsä mahdollista realisoida jollain algoritmilla tai automaatiolla. Lambdalaskennan synnyinvuosien jälkeen hiljalleen kehittyi laskettavuuden teoria, jossa tutkitaan mitä tällaisilla mekaanisilla järjestelmillä, kuten tietokoneella, voi edes periaatteessa ratkoa.

Luvussa 2 esitellään rekursiivisten funktioiden perhe sekä rekursiivisesti ratkeavat relaatiot. Aikaisempi tietämys matemaattisesta logiikasta ja rekursiivisista funktioista on hyödyksi, sillä kaikkien määritelmien semanttista oikeellisuutta ei käsitellä.

Luvussa 3 esitellään lambdalaskennan lausekkeet ja muunnossäännöt sekä todistetaan Churchin-Rosserin lause, joka kuittaa lambda-laskennan toimivaksi laskentajärjestelmäksi. Lisäksi todistetaan lambdatermien kiintopistelause, jota vastaava tulos rekursiivisille funktioille on huomattavasti mutkikkaamman todistuksen takana. Aikaisempaa tietämystä lambdalaskennasta ei edellytetä.

Luvussa 4 esitellään lambda-määriteltävien funktioiden joukko, joka viimeisessä luvussa osoitetaan samaksi rekursiivisten funktioiden joukon kanssa. Usein laskettavuuden teoriassa sallitaan osittaiset funktiot, joiden arvoa ei kaikissa pisteissä välttämättä ole määritelty. Tässä tutkielmassa käsitellään kuitenkin vain totaaleja funktioita.

Avainsanat — Nyckelord — Keywords

Laskettavuuden teoria, Matemaattinen logiikka, Rekursiiviset funktiot, Lambdakalkyyli

Säilytyspaikka — Förvaringsställe — Where deposited

Kumpulan tiedekirjasto

Muita tietoja — Övriga uppgifter — Additional information

Ohjaaja: Åsa Hirvonen

Sisältö

Luku 1

Johdanto

David Hilbertin (1862-1943) unelma oli 1920-luvulla aksiomatisoida matematiikka niin, että kaikki sen lauseet voitaisiin tarkistaa, tai jopa todistaa, jollain automaattisella menetelmällä. Kurt Gödel (1906-1978) valjasti käyttöön erään tällaisen automatisoituvan laskennan mallinnuksen, rekursiiviset funktiot, todistaakseen epätäydellisyyslauseensa. Niillä hän näytti ettei unelmasta tule mitään. Ajatus koneellisesta laskennan mallista jäi silti elämään.

Loogikko Alonzo Church (1903-1995) oli kaavaillut matematiikan perustaksi omaa laskennan formalisointia, *lambdalaskentaa* (engl. *lambda-calculus*), jonka lopullisen muodon hän julkaisi vuonna 1936. Lambdalaskenta (joka toisinaan on suomennettu myös lambdakalkyyliksi) koostuu funktioita esittävistä kaavoista, ja niille määritellyistä muunnossäännöistä.

Lambdalaskennalla määriteltävät operaatiot on osoitettu yhteneviksi sekä rekursiivisten funktioiden että useiden muiden laskentaformalismien kanssa. Tämän johdosta on päädytty otaksumaan, että rekursiiviset operaatiot ovat ne, joita on ylipäänsä mahdollista realisoida jollain algoritmilla tai automaatiolla. Vuodesta 1936 eteenpäin hiljalleen kehittyi laskettavuuden teoria (kutsutaan myös rekursioteoriaksi), jossa tutkitaan mitä tällaisilla mekaanisilla järjestelmillä, kuten tietokoneella, voi edes periaatteessa ratkoa.

Formaalina laskennan mallina lambdalaskentaa tunnetumpi lienee Alan Turingin (1912-1954) lanseeraamat Turingin koneet. Ne tottelevat hyvin määriteltyjä ohjelmointikäskyjä ja olivat mallina myöhemmin keksitylle tietokoneelle. Lambdalaskenta on kuitenkin jäänyt eloon niin kutsuttujen funktionaalisten ohjelmointikielten kautta. Ne painottavat funktioiden asemaa merkityksellisinä objekteina, pelkästään niiden tuottamien arvojen sijaan. Kuten luvussa 4 nähdään, informaatiota, kuten järjestetty pari, voidaan koodata funktioiksi sen sijaan, että sille olisi jokin erillinen tietotyyppi. Lambdalaskennasta sikisivät ohjelmointikielet LISP (*List Processing*) ja ML (*Meta Language*), joiden nykypäivän jälkeläisiä ovat muun muassa Clojure ja Haskell.

Tämän tutkielman päämäärä on osoittaa rekursiivisten funktioiden ja lambda-määriteltävien funktioiden ekvivalenssi. Luvussa 2 esitellään rekursiivisten funktioiden perhe sekä rekursiivisesti ratkeavat relaatiot. Aikaisempi tietämys matemaattisesta logiikasta ja rekursiivisista funktioista on hyödyksi, sillä kaikkien määritelmien semanttista oikeellisuutta ei käsitellä.

Luvussa 3 esitellään lambdalaskennan lausekkeet ja muunnossäännöt. Lisäksi todistetaan Churchin-Rosserin lause, joka kuittaa lambda-laskennan toimivaksi laskentajärjestelmäksi. Todistus mukailee Henk Barendregtin kirjasta [1] löytyvää kuvallista todistusta. Kuvat on toteutettu GeoGebra-sovelluksella [3]. Aikaisempaa tietämystä lambdalaskennasta ei edellytetä.

Luvussa 4 esitellään lambda-määriteltävien funktioiden joukko, jossa on ne luonnollisten lukujen funktiot, jotka voidaan koodata lambdalaskennan kaavoina. Tämä joukko viimeisessä luvussa osoitetaan samaksi rekursiivisten funktioiden joukon kanssa. Usein laskettavuuden teoriassa sallitaan osittaiset funktiot, joiden arvoa ei kaikissa pisteissä välttämättä ole määritelty. Tässä tutkielmassa käsitellään kuitenkin vain totaaleja funktioita. Tämä rajoitus johtuu siitä, että lambdalaskennan funktioiden 'määrittelemättömien' arvojen määrittelyyn liittyy monimutkaisuuksia ja polemiikkia.

Ensimmäisenä rekursiivisten ja lambda-määriteltävien funktioiden yhtenevyyden todisti Churchin oppilas Stephen Cole Kleene (1909-1994) vuonna 1936 artikkelissaan *λ-definability and recursiveness* ([5]).

Luku 2

Rekursiivisten funktioiden perhe

Rekursio tarkoittaa määrittelyä induktiivisesti – tämän tutkielman tapauksessa funktion arvon määrittelyä aikaisempien arvojen avulla. Perusluonteinen esimerkki tästä on kenties yksi monipuolisimmin karakterisoiduista funktioista matematiikassa:

$$\begin{cases} 0! = 1 \\ (n+1)! = n!(n+1). \end{cases}$$

Yleisesti kyseistä muotoa kutsutaan *primitiivirekursioksi.* Pidättäydymme funktioissa luonnollisilta luvuilta luonnollisille luvuille. Olkoon $a \in \mathbb{N}$ ja $g \colon \mathbb{N}^2 \to \mathbb{N}$. Funktio $h \colon \mathbb{N} \to \mathbb{N}$ on saatu primitiivirekursiolla, jos se toteuttaa yhtälöt

$$\begin{cases} h(0) = a \\ h(y+1) = g(y, h(y)). \end{cases}$$

Yleisempi määritelmä, joka mahdollistaa useiden parametrien käsittelyn on seuraava. Tässä tutkielmassa n on aina mielivaltainen luonnollinen luku, ja $\boldsymbol{x} = (x_1, ..., x_n)$ on mielivaltainen joukon $\mathbb{N}^n$ alkio, jollei muuta mainita. Olkoon $f \colon \mathbb{N}^n \to \mathbb{N}$ ja $g \colon \mathbb{N}^{n+2} \to \mathbb{N}$. Jos h on muotoa

$$\begin{cases} h(\boldsymbol{x}, 0) = f(\boldsymbol{x}) \\ h(\boldsymbol{x}, y+1) = g(\boldsymbol{x}, y, h(\boldsymbol{x}, y)), \end{cases}$$

sanotaan, että h on saatu primitiivirekursiolla funktioista f ja g.

Niin kutsuttu primitiivirekursiivisten funktioiden joukko kattaa laajan paletin erilaisia funktioita, mutta laajempi perhe, *rekursiiviset funktiot*, tarvitsee lisäksi eräänlaisen hakuoperaattorin. Rekursiivisten funktioiden teoriassa toinen keskeinen käsite on *minimalisointi.*

Määritelmä 2.1. Olkoon $f\colon \mathbb{N}^{n+1} \to \mathbb{N}$. Minimalisointi määritellään *μ-operaattorilla*:

$$\mu y(f(\boldsymbol{x}, y) = 0) = \text{'pienin } y \text{ jolla } f(\boldsymbol{x}, y) = 0\text{'}.$$

Huomautus 2.2. Jos halutaan minimalisoinnin olevan totaalinen operaatio, on tehtävä oletus, että tällainen 'pienin y' on olemassa.

Minimalisointi on hieman samankaltainen toimenpide kuin primitiivirekursio. Rekursiivisesti voidaan määritellä $\mu y(f(\boldsymbol{x}, y) = 0) = g(\boldsymbol{x}, 0)$, missä

$$g(\boldsymbol{x}, y) = \begin{cases} y, & \text{jos} \quad f(\boldsymbol{x}, y) = 0 \\ g(\boldsymbol{x}, y+1), & \text{muuten.} \end{cases}$$

Operaattori μ siis aloittaa nollasta ja kapuaa ylöspäin kunnes se löytää funktion f nollakohdan. Perimmäinen ero primitiivirekursioon (ja syy miksi minimalisointia ei voi toteuttaa primitiivirekursion avulla) on se, ettemme tiedä milloin prosessi päättyy: primitiivirekursiivinen funktio lopettaa laskennan, kun y:n arvo on pudonnut nollaan ($h(\boldsymbol{x}, 0)$), mutta minimalisaation laskenta voi jatkua mielivaltaisen pitkään.

Rekursiiviset funktiot määritellään induktiivisesti. Ensin asetetaan niin sanotut perusfunktiot (engl. *initial functions*), ja kaikki näille tehdyt yhdisteet, primitiivirekursiot ja minimalisoinnit säilyttävät rekursiivisuuden.

Määritelmä 2.3. *Rekursiiviset* funktiot $h\colon \mathbb{N}^n \to \mathbb{N}$ määritellään seuraavasti:

1. Vakiofunktio nollalle $Z(n) = 0$ on rekursiivinen.

2. Seuraajafunktio $S(n) = n + 1$ on rekursiivinen.

3. Projektiofunktio $P_i^n(x_1, ..., x_n) = x_i$, missä $1 \leq i \leq n$, on rekursiivinen.

4. Jos $f\colon \mathbb{N}^n \to \mathbb{N}$ ja $g_i\colon \mathbb{N}^m \to \mathbb{N}$ ovat rekursiivisia, kun $1 \leq i \leq n$, niin yhdistetty funktio
$$h(x_1, ..., x_m) = f(g_1(x_1, ..., x_m), ..., g_n(x_1, ..., x_m))$$
on rekursiivinen.

5. Jos $f\colon \mathbb{N}^n \to \mathbb{N}$ ja $g_i\colon \mathbb{N}^{n+2} \to \mathbb{N}$ ovat rekursiivisia, niin primitiivirekursiolla saatu funktio
$$\begin{cases} h(\boldsymbol{x}, 0) = f(\boldsymbol{x}) \\ h(\boldsymbol{x}, y+1) = g(\boldsymbol{x}, y, h(\boldsymbol{x}, y)) \end{cases}$$
on rekursiivinen.

6. Jos $f\colon \mathbb{N}^{n+1} \to \mathbb{N}$ on rekursiivinen, ja kaikilla $\boldsymbol{x} \in \mathbb{N}^n$ on olemassa $y \in \mathbb{N}$ jolla $f(\boldsymbol{x}, y) = 0$, niin minimalisoinnilla saatu funktio

$$h(\boldsymbol{x}) = \mu y(f(\boldsymbol{x}, y) = 0)$$

on rekursiivinen.

Huomautus 2.4. Kyseinen määritelmä (erityisesti kohta 6) koskee vain totaalisia funktioita. Tätä luokkaa kutsutaan myös μ-rekursiivisiksi funktioiksi. Usein laskennan teoriassa käytetty funktioperhe on osittaisten rekursiivisten funktioiden joukko (engl. *partial recursive functions*), joka sallii sen etteivät funktiot ole kaikkialla määriteltyjä. Tämä tutkielma on kuitenkin rajoitettu totaalisiin funktioihin teknisten yksityiskohtien vähentämiseksi.

Rekursiivisia funktioita kutsutaan myös laskettaviksi funktioiksi. Niin sanottu *Churchin teesi* on laajalti hyväksytty väittämä, että rekursiiviset funktiot ovat täsmälleen ne funktiot, jotka voidaan laskea jollain äärellisellä mekanistisella algoritmilla. Tämä funktioperhe siis kuvaa operaatioita, jotka tietokone (ilman aika- ja tilarajoituksia) pystyy toteuttamaan.

Laskettavuuden teoriassa on perinteistä rajoittautua luonnollisiin lukuihin yksinkertaisuuden vuoksi. Luonnolliset luvut edustavat diskreettiä ääretöntä joukkoa, ja niin halutessaan matemaatikko voi niistä konstruoida esimerkiksi kokonaisluvut.

Lause 2.5. *Seuraavat funktiot ovat rekursiivisia.*

i) $a + b$

Todistus. Summa saadaan primitiivirekursiolla rekursiivisista funktioista P_1^1 ja S:

$$\begin{cases} a + 0 = a = P_1^1(a) \\ a + (b + 1) = (a + b) + 1 = S(a + b). \end{cases}$$

□

ii) $a \cdot b$

Todistus. Tulo saadaan primitiivirekursiolla rekursiivisista funktioista Z ja $a+b$:

$$\begin{cases} a \cdot 0 = 0 = Z(a) \\ a \cdot (b + 1) = (a \cdot b) + a. \end{cases}$$

□

iii) a^b

Todistus. $\begin{cases} a^0 = 1 = S(Z(a)) \\ a^{b+1} = a^b \cdot a. \end{cases}$ □

iv) $a \dot{-} 1 = \begin{cases} 0, & \textit{jos } a = 0 \\ a - 1, & \textit{muuten} \end{cases}$

Todistus. $\begin{cases} 0 \dot{-} 1 = 0 \\ (a+1) \dot{-} 1 = a. \end{cases}$ □

v) $a \dot{-} b = \begin{cases} 0, & \textit{jos } a \leq b \\ a - b, & \textit{muuten} \end{cases}$

Todistus. $\begin{cases} a \dot{-} 0 = a \\ a \dot{-} (b+1) = (a \dot{-} b) \dot{-} 1. \end{cases}$ □

Lause 2.6. *Jos* $f\colon \mathbb{N}^{n+1} \to \mathbb{N}$ *on rekursiivinen, niin funktio* $g(\boldsymbol{x}, y) = \prod_{z \leq y} f(\boldsymbol{x}, z)$ *on rekursiivinen.*

Todistus. $\begin{cases} g(\boldsymbol{x}, 0) = f(\boldsymbol{x}, 0) \\ g(\boldsymbol{x}, y+1) = g(\boldsymbol{x}, y) \cdot f(\boldsymbol{x}, y+1). \end{cases}$ □

2.1 Rekursiiviset relaatiot

Relaatiot kuvaavat luonnollisten lukujen ominaisuuksia tai niiden välisiä suhteita. Rekursiivinen relaatio on sellainen, että luvun tai lukujonon kuulumisen relaatioon voi tarkistaa rekursiivisella funktiolla. Churchin teesin vuoksi rekursiivisia relaatioita (tai 'predikaatteja') kutsutaan myös ratkeaviksi.

Määritelmä 2.7. Relaatio $R \subseteq \mathbb{N}^n$ on rekursiivinen, jos sen karakteristinen funktio

$$c_R(\boldsymbol{x}) = \begin{cases} 1, & \text{jos} \quad \boldsymbol{x} \in R \\ 0, & \text{jos} \quad \boldsymbol{x} \notin R \end{cases}$$

on rekursiivinen. Relaatioon 'kuulumista' $\boldsymbol{x} \in R$ tullaan kirjoittamaan pääasiallisesti $R(\boldsymbol{x})$.

Esimerkki 2.8. Relaatio $'x \leq y'$ on rekursiivinen, sillä voidaan määritellä karakteristinen funktio

$$c_{\leq}(x, y) = 1 \dot{-} (x \dot{-} y).$$

Huomautus 2.9. Rekursiivisen relaation määritelmästä seuraa välittömästi, että relaatio $R'(x, y) \equiv R(f(x), g(y))$ on rekursiivinen, jos relaatio R sekä funktiot f ja g ovat rekursiivisia. Täten esimerkiksi relaatio $R'(x, y) \equiv {}'x \leq y + 1'$ on rekursiivinen.

Tässä tutkielmassa käytetään ylläolevaa merkintää '≡' relaatioiden määrittelyyn.

Määritelmä 2.10. Olkoon relaatiot $R_1, R_2 \subseteq \mathbb{N}^n$ rekursiivisia. Määritellään seuraavat rekursiiviset relaatiot:

- $\neg R_1(\boldsymbol{x})$ määritellään karakteristisella funktiolla

$$c_A(\boldsymbol{x}) = 1 \dot{-} c_{R_1}(\boldsymbol{x}).$$

- $[R_1(\boldsymbol{x})$ ja $R_2(\boldsymbol{x})]$ määritellään karakteristisella funktiolla

$$c_B(\boldsymbol{x}) = c_{R_1}(\boldsymbol{x}) \cdot c_{R_2}(\boldsymbol{x}).$$

- $[R_1(\boldsymbol{x})$ tai $R_2(\boldsymbol{x})]$ määritellään karakteristisella funktiolla

$$c_C(\boldsymbol{x}) = c_{R_1}(\boldsymbol{x}) + c_{R_2}(\boldsymbol{x}) - c_{R_1}(\boldsymbol{x}) \cdot c_{R_2}(\boldsymbol{x}).$$

- $[R_1(\boldsymbol{x}) \to R_2(\boldsymbol{x})] \equiv [\neg R_1(\boldsymbol{x})$ tai $R_2(\boldsymbol{x})]$.

Esimerkki 2.11. Relaatio $'x = y'$ on rekursiivinen, sillä

$$'x = y' \equiv \big[x \leq y \text{ ja } y \leq x\big].$$

Rajoitettu kvantifiointi säilyttää relaation rekursiivisuuden. Rajoittamattoman kvantifioinnin kohdalla näin ei kuitenkaan ole. Tarvittaisiin rajattoman monta laskenta-askelta, jotta voitaisiin varmistaa päteekö jokin tosi väite kaikilla luonnollisilla luvuilla.

Määritelmä 2.12. Olkoon relaatio $R \subseteq \mathbb{N}^{n+1}$ rekursiivinen. Määritellään seuraavat rekursiiviset relaatiot:

- $(\forall y \leq x_1)\big[R(\boldsymbol{x}, y)\big]$ määritellään karakteristisella funktiolla

$$c_D(\boldsymbol{x}) = \prod_{y \leq x_1} c_R(\boldsymbol{x}, y).$$

- $(\exists y \leq x_1)\big[R(\boldsymbol{x}, y)\big] \equiv \neg(\forall y \leq x_1)\big[\neg R(\boldsymbol{x}, y)\big]$.

Relaatioiden hyöty on tulevan teorian kannalta se, että niiden avulla voidaan esittää funktioita minimalisoinnilla lukuteoreettisten ominaisuuksien suhteen: 'pienin y jolla R pätee' on saatavissa rekursiivisesti, jos R on rekursiivinen.

Määritelmä 2.13. Olkoon $R \subseteq N^{n+1}$ relaatio. Määritellään funktio

$$\mu y\big[R(\boldsymbol{x}, y)\big] = \mu y\big(c_{\neg R}(\boldsymbol{x}, y) = 0\big).$$

Määritelmästä seuraa suoraan, että kyseinen funktio on rekursiivinen aina kun R on rekursiivinen. Esitettyjen määritelmien avulla voidaan esittää lukuteorian ominaisuuksia rekursiivisesti.

Lause 2.14. *Seuraavat relaatiot ja funktiot ovat rekursiivisia.*

1. $x|y \equiv (\exists z \leq y)\big[y = x \cdot z\big]$.
   Luku y on jaollinen luvulla x.

2. $Prim(x) \equiv x > 1$ ja $\neg(\exists z \leq x)\big[z \neq 1$ ja $z \neq x$ ja $z|x\big]$.
   Luku x on alkuluku.

3. $\begin{cases} p_0 = 0 \\ p_{x+1} = \mu y\big[y > p_x \text{ ja } Prim(y)\big]. \end{cases}$
   Funktio $x \mapsto p_x$ antaa x:nnen alkuluvun.

Luku 3

Johdatus lambdalaskentaan

David Hilbertin ratkeavuusongelma (*Entscheidungsproblem*) kysyy onko olemassa algoritmia joka tarkistaa, onko mikä tahansa syötetty ensimmäisen kertaluvun predikaattilogiikan lause validi vai ei. Alonzo Church oli vakuuttunut, ettei tällaista algoritmia ole, mutta tämän todistamiseen hän tarvitsi formaalin määritelmän algoritmille. Hän kehitti lambdalaskennan (engl. *lambda-calculus*) 1920- ja 1930-luvuilla tätä tarkoitusta varten ja todisti vuonna 1936, että tällä formalismilla algoritmi on mahdoton.

Lambdalaskennan lähtökohtana on joukko-opin sijasta käsite operaatiosta. Ideana on esittää merkkijonojen avulla funktioita, joiden arvot lasketaan puhtaasti syntaktisilla muunnoksilla. Abstraktiosymbolilla λ sidotaan jokin muuttuja, kuten x, ja määritellään laskettava askel $x \mapsto f(x)$ kirjoittamalla $\lambda x.f(x)$. Tällaista funktiota sovelletaan kirjoittamalla parametri funktion perään. Esimerkiksi sovellus $(\lambda x.x^2)2$ saisi arvon 4.

Käytettävässä kielessä on seuraavat merkit:

- muuttujat $x_0, x_1, x_2, ...$
- abstraktiosymbolit 'λ' ja '.'
- sulkeet '(' ja ')'.

Pelkistetystä syntaksista huolimatta tällä formalismilla voidaan ilmaista kaikki rekursiiviset funktiot, kuten tämä tutkielma osoittaa. Keskeisimmät rakenteet lambdalaskennan kaavoissa ovat funktion *soveltaminen* sekä funktioiden konstruointi *abstrahoimalla.*

Määritelmä 3.1. *Lambdatermit* määritellään seuraavasti:

1. Muuttujat x_i ovat lambdatermejä kaikilla $i \in \mathbb{N}$.
2. Jos M ja N ovat lambdatermejä, niin termin M *sovellus* termiin N, (MN), on lambdatermi.

3. Jos M on lambdatermi ja x_i on muuttuja, niin *abstraktio* $(\lambda x_i.M)$ suhteessa muuttujaan x_i on lambdatermi.

Määritelmä 3.2. Termin *vapaat* muuttujat määritellään seuraavasti:

1. Muuttuja x_i on vapaa termissä x_i.
2. Termin (MN) vapaat muuttujat ovat termien M ja N vapaat muuttujat.
3. Termin $(\lambda x_i.M)$ vapaat muuttujat ovat termin M vapaat muuttujat lukuunottamatta muuttujaa x_i.

Jos muuttuja ei ole vapaa, sitä kutsutaan *sidotuksi.*

Huomaa, että muuttujat eivät kuulu mihinkään määrittelyjoukkoon, eikä niillä yksinään ole mitään tulkintaa: x_1 ja (x_1x_2) ovat päteviä lambdatermejä, mutta käytännössä useimmiten kiinnostuksemme kohdistuu vain sellaisiin termeihin, joissa ei ole vapaita muuttujia.

Tästä eteenpäin M ja N ovat aina mielivaltaisia lambdatermejä ellei muuta mainita. Määritelmän mukaisille lambdatermeille tulee myös käyttöä, mutta otetaan seuraavat kirjoituskonventiot käyttöön tekstin siistimiseksi.

Muuttujina tullaan käyttämään merkkien x_i sijasta myös muita kirjaimia. Sulut jätetään pois, jos se ei aiheuta epäselvyyttä. Ylimääräisten sulkujen välttämiseksi sovitaan, että sovellukset luetaan vasemmalta oikealle:

$$\begin{aligned} M_1M_2M_3 &\quad \text{tarkoittaa} \quad ((M_1M_2)M_3), \text{ ja yleisemmin} \\ M_1M_2...M_n &\quad \text{tarkoittaa} \quad (\cdot\cdot((M_1M_2)M_3)\cdots M_n). \end{aligned} \tag{3.3}$$

Lambdalaskenta mahdollistaa useiden muuttujien funktioiden esittämisen yhden muuttujan funktioiden ketjuna. Esimerkiksi jos kahden muuttujan funktiota $f(x, y)$ halutaan esittää lambdaterminä $(\lambda xy.Fxy)$, parametrit x ja y voidaan käsitellä peräjälkeen: ylläolevan konvention mukaan sovellus Fxy tulkitaan $(Fx)y$. Vaikka funktiota F ajateltaisiin kahden muuttujan funktiona, niin sovellus (Fx) palauttaa uuden funktion, missä x on kiinnitetty. Tätä kutsutaan englanninkielessä nimellä *currying* loogikko Haskell Curryn mukaan.

$$\begin{aligned} \lambda x_1x_2.M &\quad \text{tarkoittaa} \quad \lambda x_1.(\lambda x_2.M), \text{ ja yleisemmin} \\ \lambda x_1...x_n.M &\quad \text{tarkoittaa} \quad \lambda x_1.(\lambda x_2.(\cdots(\lambda x_n.M)\cdot\cdot)). \end{aligned}$$

Esimerkki 3.4. Identtinen funktio voidaan esittää lambdaterminä $\lambda x.x$, jonka soveltamista termiin M merkitään $(\lambda x.x)M$.

On tärkeää huomata, ettei termin M olemukseen oteta kantaa: se voi yhtä hyvin olla muuttuja kuin funktiokin. Lambdalaskennassa ei ole erittelyä funktioiden ja parametrien välillä. Myös

$$(\lambda x.x)(\lambda x.x)$$

on pätevä lambdatermi. Siinä identtistä funktiota sovelletaan parametriin, joka itsekin on identtinen funktio. Matematiikassa tyypillistä on ajatella funktioiden olevan korkeamman tason objekteja kuin parametrinsa. Tässä kuitenkin käsitetään funktio toimenpiteenä merkkijonoille (lambdatermeille), ja sen sijaan, että lähtö- ja maalijoukot olisivat ennalta annettuja, funktiota sovelletaan sellaisiin termeihin, jotka tilanteessa ovat tarkoituksenmukaisia.

Esimerkki 3.5. Termi $\lambda x.(\lambda y.y)$ on funktio, joka palauttaa parametrista huolimatta identtisen funktion. Tämä voitaisiin *currying*-merkinnällä kirjoittaa myös $\lambda xy.y$, mikä näyttää kahden parametrin funktiolta, tarkemmin sanoen projektiolta. Muista kuitenkin, että 'usean parametrin' funktiot ovat todellisuudessa vain funktioita, jotka yhdelle parametrille palauttavat uuden funktion.

Käytetään konventiota, että ilman sulkuja λ-abstraktio ulottuu lausekkeen loppuu:

$$\lambda x.MN \quad \text{tarkoittaa} \quad \lambda x.(MN),$$

eli sovellus kirjoitettaisiin sulkujen kanssa $(\lambda x.M)N$.

Esimerkki 3.6. Funktio

$$\Delta = \lambda x.xx$$

aiheuttaa parametrin soveltamisen itseensä.

Funktiosovellusten arvo voidaan määrittää kahdella muunnossäännöllä. Termit sievenevät niin sanotulla beetareduktiolla. Esimerkiksi $(\lambda x.xx)N$ sievenee termiksi NN, kun N sijoitetaan muuttujan x paikalle termissä xx. Käytetään merkintää $M[x := N]$ termille M, jossa muuttujan x vapaiden esiintymien tilalle on sijoitettu termi N. Lupaava sääntö olisi

$$(\lambda x.M)N \to M[x := N].$$

Toisinaan sattuu kuitenkin muuttujanimien yhteentörmäyksiä. Muunnos

$$(\lambda x.(\lambda y.xy))y \to \lambda y.yy \tag{3.7}$$

sekoittaa muuttujan y kaksi eri käyttötapausta eikä säilytä funktion tarkoitusta. Beetareduktio on sallittu vain, jos parametrin vapaat muuttujat säilyvät sijoituksen yhteydessä vapaina. Tästä syystä on hyödyllistä on käyttää seuraavaa sopimusta.

Konventio 3.8 (Muuttujaoletus)**.** *Oletetaan, että kaikkien lauseissa, määritelmissä ja esimerkeissä esiintyvien termien sidotut muuttujat ovat erillisiä termeissä esiintyvistä vapaista muuttujista.*

Tämä on perusteltua sillä, että – muuttujien lukumäärän ollessa rajaton – sidottu muuttuja voidaan aina valita joksikin muista erilliseksi muuttujaksi. Muunnossääntöjä varten määritellään juuri esitelty sijoitusoperaattori täsmällisesti.

Määritelmä 3.9 (Sijoitus vapaaseen muuttujaan)**.** Määritellään termi $M[x := N]$ induktiivisesti termin M rakenteen suhteen.

1. Termi M on joko muuttuja x tai jokin muu muuttuja y. Ensimmäisessä tapauksessa $x[x := N] = N$. Toisessa tapauksessa $y[x := N] = y$.
2. $(M_1M_2)[x := N] = \big(M_1[x := N]\big)\big(M_2[x := N]\big)$.
3. $(\lambda y.M_1)[x := N] = \lambda y.(M_1[x := N])$. Huomaa, että muuttujaoletuksen (3.8) perusteella y ei esiinny vapaana termissä N ja $y \neq x$.

Määritelmä 3.10. Termin *alitermit* määritellään seuraavasti:

1. Termi M on itsensä alitermi.
2. Termien M ja N alitermit ovat termin (MN) alitermejä.
3. Termin M alitermit ovat termin $(\lambda x.M)$ alitermejä.

Määritelmä 3.11. Seuraavia muunnossääntöjä voidaan soveltaa alitermeihin:

i) **Alfamuunnos.** Sidottuja muuttujia voidaan nimetä uudelleen:

$$\lambda x.M \overset{\alpha}{\to} \lambda y.M[x := y].$$

Huomaa, että muuttujaoletuksen perusteella y ei esiinny vapaana termissä M.

ii) **Beetareduktio.** Funktiota $(\lambda x.M)$ voidaan soveltaa parametriin N:

$$(\lambda x.M)N \overset{\beta}{\to} M[x := N].$$

Huomaa, että muuttujaoletuksen perusteella termissä M ei ole muuttujaa, joka voisi sijoituksen yhteydessä sitoa jonkin termin N vapaan muuttujan.

Jos termi M voidaan muuntaa termiksi N äärellisellä määrällä alfamuunnoksia ja beetareduktioita, sanotaan, että termi M on *redusoituva* termiin N ja kirjoitetaan $M \geq_\beta N$. Jos termit M ja N voidaan muuntaa samaksi termiksi äärellisellä määrällä alfamuunnoksia ja beetareduktioita, niin sanotaan, että termit M ja N ovat *beetayhtäsuuret* ja kirjoitetaan $M \overset{\beta}{=} N$.

Huomautus 3.12. Beetareduktio on yksisuuntainen. Vaikka $M \overset{\beta}{\to} N$ pätisi, niin emme välttämättä saa reduktiota $N \overset{\beta}{\to} M$.

Määritelmä 3.13. Termi on *normaalimuodossa*, jos sen aliterminä ei ole yhtään λ-abstraktion sovellusta (ja täten sitä ei voi enää beetaredusoida). Sanotaan, että termillä M on normaalimuoto, jos se on redusoituva johonkin normaalimuotoiseen termiin M'.

Esimerkki 3.14. Nyt aikaisemmin ongelmallinen lauseke (3.7) saadaan sievennettyä oikein:

$$(\lambda x.(\lambda y.xy))y \overset{\alpha}{\to} (\lambda x.(\lambda z.xz))y \overset{\beta}{\to} \lambda z.yz,$$

joten $(\lambda x.(\lambda y.xy))y \overset{\beta}{=} \lambda z.yz$.

Esimerkin (3.6) funktiolle $\Delta = \lambda x.xx$ pätee

$$\Delta M \overset{\beta}{=} MM.$$

Lambdalaskennan heikkouksiin kuuluu se, ettei kaikki termit ole sievennettävissä. Sovellettaessa itseensä funktio Δ aiheuttaa beetareduktiolla kehän

$$\Delta\Delta \overset{\beta}{\to} \Delta\Delta.$$

Vielä rumemmin käyttäytyy $\Delta_3 = \lambda x.xxx$, kuten nähdään seuraavasta: muista, että merkinnän (3.3) mukaan termi xxx tulkitaan $(xx)x$.

$$\Delta_3\Delta_3 \overset{\beta}{\to} \Delta_3\Delta_3\Delta_3 \overset{\beta}{\to} \Delta_3\Delta_3\Delta_3\Delta_3 \overset{\beta}{\to} ...$$

Kaikille termeille ei siis ole normaalimuotoa. Tämä on jokseenkin analogista Turingin koneeseen, joka ei pysähdy, tai osittaiseen funktioon, jonka arvoa ei ole määritelty.

3.1 Churchin-Rosserin lause

Sievennys normaalimuotoon voi tapahtua useaa reittiä pitkin. Jos v ja u ovat vapaita muuttujia, molemmat seuraavista reduktioista ovat päteviä:

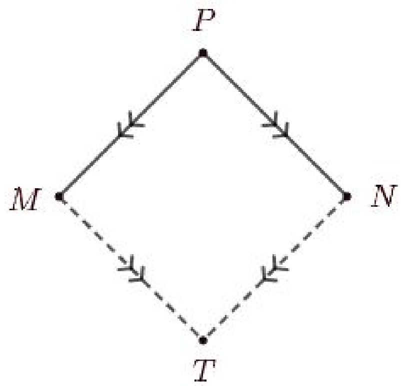
a) $(\lambda y.(\lambda x.x)v)u \stackrel{\beta}{\to} (\lambda x.x)v \stackrel{\beta}{\to} v,$ ja

b) $(\lambda y.(\lambda x.x)v)u \stackrel{\beta}{\to} (\lambda y.v)u \stackrel{\beta}{\to} v.$

Tässä tapauksessa molemmat sievennykset päätyvät lopulta samaan normaalimuotoon. Ei ole itsestäänselvää, että kaikkien lambdatermien laita on näin. Lambdalaskennan kannalta kriittinen Churchin-Rosserin lause osoittaa, että mikään lambdatermi ei voi sieventyä kahteen eri normaalimuotoon. Tässä kappaleessa ajatellaan alfamuunnoksen piirissä olevia lambdatermejä identtisinä ja kirjoitetaan $M \stackrel{\alpha}{=} N$, jos N ja M voidaan muuntaa samoiksi termeiksi alfamuunnosten avulla.

Lause 3.15 (Churchin-Rosserin lause)**.** *Olkoon P, M ja N lambdatermejä. Jos P on redusoituva termeihin M ja N, niin on olemassa termi T, johon sekä M että N ovat redusoituvia.*

Kuva 3.1: Churchin-Rosserin lause



Kuvassa 3.1 on lauseen havainnollistus: yhtenäiset nuolet kuvaavat oletettuja redusoituvuuksia. Churchin-Rosserin lauseen mukaan katkoviivalla kuvatut reduktiot ovat aina olemassa. Ennen lauseen todistusta pannaan merkille sen kaksi merkittävää korollaaria.

Korollaari 3.16. *Jos termillä P on normaalimuoto, niin se on yksikäsitteinen.*

Todistus. Huomaa että, jos termi Q on normaalimuodossa ja $Q \geq_\beta Q'$, niin $Q \stackrel{\alpha}{=} Q'$, koska termiä Q ei voi beetaredusoida. Oletetaan, että termi P on redusoituva joihinkin normaalimuotoisiin termeihin M ja N. Churchin-Rosserin lauseen nojalla on olemassa T, jolla $M \geq_\beta T$ ja $N \geq_\beta T$. Ylläolevan huomion perusteella $M \stackrel{\alpha}{=} T \stackrel{\alpha}{=} N$. $\square$

Korollaari 3.17. *Beetayhtäsuuruus ei ole triviaali: jos termeillä M ja N on erilliset normaalimuodot, ne eivät ole keskenään beetayhtäsuuret.*

Todistus. Oletetaan, että termit M ja N redusoituvat erillisiin normaalimuotoihin M' ja N'. Jos $M \overset{\beta}{=} N$, niin on olemassa termi Q, johon molemmat niistä sievenevät. Koska termi M redusoituu termeihin Q ja M', niin Churchin-Rosserin lauseen nojalla on olemassa termi $T \overset{\alpha}{=} M'$, johon myös Q redusoituu. Nyt termi N redusoituu sekä normaalimuotoon N' että (termin Q kautta) normaalimuotoon M', mikä on ristiriita edellisen korollaarin (3.16) kanssa. $\square$

Tämän kappaleen tavoitteena on todistaa Churchin-Rosserin lause. Todistukseen tarvitaan avuksi liuskalemma (engl. *Strip lemma*), joka on Churchin-Rosserin lauseen erityistapaus: redusoituvuuden $P \geq_\beta M$ sijaan oletetaan beetareduktio $P \overset{\beta}{\to} M$. Liuskalemman todistus seuraa Henk Barendregtin menetelmää, jossa käytetään pian määriteltävää lambdalaskennan laajennosta. Tässä laajennoksessa on eräänlainenlainen kirjanpito, jossa symbolilla '$\tilde{\lambda}$' korvamerkataan yksittäinen beetareduktioaskel. Intuitiivisesti strategia on se, että reduktiossa $P \overset{\beta}{\to} M$ redusoituva lambda laitetaan muistiin, ja termi T saadaan kun termistä N redusoidaan tämä sama lambda.

Määritelmä 3.18. *Laajennettu lambdalaskenta* määritellään seuraavilla kohdilla:

- Laajennetun lambdalaskennan termit määritellään samoin kuin lambdatermit, mutta lisäyksellä

  4. Jos M ja N ovat termejä sekä x on muuttuja, niin $((\tilde{\lambda}x.M)N)$ on termi.

  Huomaa, että aaltoviivalla merkitään pelkästään sovelluksia, ei soveltamattomia abstraktioita.

- Laajennetun lambdalaskennan termien vapaat muuttujat on määritelty samoin, kuin lambdatermien vapaat muuttujat, lisäyksellä

  4. Termin $((\tilde{\lambda}x.M)N)$ vapaat muuttujat ovat termin M vapaat muuttujat lukuunottamatta muuttujaa x sekä termin N vapaat muuttujat.

- Sijoitus laajennetun lambdalaskennan termin vapaaseen muuttujaan määritellään kuten lambdalaskennan sijoitus, lisäyksellä

  4. $((\tilde{\lambda}x.M)N)[x := P] = \big((\tilde{\lambda}x.M[x := P])(N[x := P])\big)$.

- Laajennetun lambdalaskennan alitermit on määritelty samoin kuin lambdalaskennan alitermit, lisäyksellä

4. Termien M ja N alitermit ovat termin $((\tilde{\lambda}x.M)N)$ alitermejä.

- Laajennetussa lambdalaskennassa määritellään alitermien muunnossäännöt seuraavasti:

  i) **Laajennettu alfamuunnos.**

  $$\lambda x.M \xrightarrow{\alpha} \lambda y.M[x := y], \text{ ja } (\tilde{\lambda}x.M)N \xrightarrow{\alpha} (\tilde{\lambda}y.M[x := y])N.$$

  ii) **Laajennettu beetareduktio.**

  $$(\lambda x.M)N \xrightarrow{\beta} M[x := N], \text{ ja } (\tilde{\lambda}x.M)N \xrightarrow{\beta} M[x := N].$$

  Jos termi M voidaan muuntaa termiksi N äärellisellä määrällä laajennettuja alfamuunnoksia ja beetareduktioita, kirjoitetaan $M \geq_{\tilde{\beta}} N$.

Reduktiossa $M \geq_{\tilde{\beta}} N$ voi siis olla sekä tavallisten että aaltoviivattujen lambdojen beetareduktioita. Symbolia $\tilde{\beta}$ käytetään vain alleviivaamaan, että termit M ja N ovat laajennetun lambdalaskennan termejä.

Määritelmästä huolimatta liuskalemman todistukseen tarvitaan vain yhtä aaltoviivattua lambdaa termissä kerrallaan. Merkitään kaikkien laajennetun lambdalaskennan termien joukkoa $\tilde{\Lambda}$ ja kaikkien 'tavallisten' lambdatermien joukkoa Λ.

Määritelmä 3.19. Määritellään funktiot $|\,|\colon \tilde{\Lambda} \to \Lambda$ ja $\varphi\colon \tilde{\Lambda} \to \Lambda$ seuraavasti:

i) $|M|$ on lambdatermi, joka saadaan laajennetun lambdalaskennan termistä M, kun kaikki merkit $\tilde{\lambda}$ vaihdetaan merkkeihin λ.

ii) Funktio φ määritellään induktiivisesti laajennetun lambalaskennan termin rakenteen suhteen:

1. $\varphi(x) = x$.
2. $\varphi(MN) = \varphi(M)\varphi(N)$, jos M ei ole muotoa $\tilde{\lambda}x.M'$.
3. $\varphi(\lambda x.M) = \lambda x.\varphi(M)$.
4. $\varphi((\tilde{\lambda}x.M)N) = \varphi(M)[x := \varphi(N)]$.

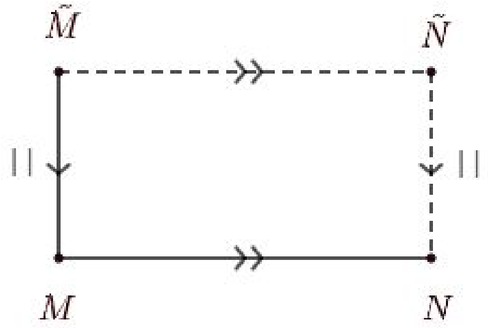
Sanalla sanoen funktio φ beetaredusoi kaikki $\tilde{\lambda}$-sovellukset ja jättää termin muuten samanlaiseksi.

Liuskalemmaa varten tarvitaan kolme apulausetta lambdatermien ja laajennetun lambdalaskennan termien suhteesta. Kuvia käytetään tulosten havainnollistamiseksi. Yhtenäiset nuolet kuvaavat oletettuja, ja katkonaiset nuolet todistettavia suhteita. Kaksipäinen nuoli kuvaa redusoituvuuksia $M \geq_\beta N$ ja $\tilde{M} \geq_{\tilde{\beta}} \tilde{N}$. Merkitään kuviin nuolet

$$\begin{aligned}
M \overset{\beta}{\to} N \quad & \text{jos} \quad M \overset{\alpha}{=} M' \overset{\beta}{\to} N \text{ jollakin } M' \in \Lambda, \\
\tilde{M} \overset{||}{\to} M \quad & \text{jos} \quad |\tilde{M}| \overset{\alpha}{=} M, \text{ ja} \\
\tilde{M} \overset{\varphi}{\to} M \quad & \text{jos} \quad \varphi(\tilde{M}) \overset{\alpha}{=} M.
\end{aligned}$$

Lemma 3.20. *Olkoon $\tilde{M}$ laajennetun lambdalaskennan termi ja $M \overset{\alpha}{=} |\tilde{M}|$. Jos M on redusoituva johonkin lambdatermiin N, niin $\tilde{M}$ on redusoituva sellaiseen termiin $\tilde{N} \in \tilde{\Lambda}$, jolla $|\tilde{N}| \overset{\alpha}{=} N$ (ks. kuva 3.2).*

Kuva 3.2: Lemma 3.20

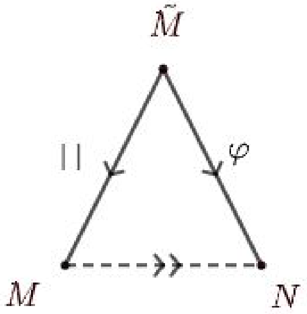


Todistus. Termit $\tilde{M}$ ja M ovat muuten identtisiä, paitsi termissä $\tilde{M}$ saattaa esiintyä aaltoviivoja lambdojen yllä. Termi $\tilde{N}$ saadaan kun termistä $\tilde{M}$ beetaredusoidaan kaikki vastaavat sovellukset (aaltoviivoilla tai ilman) kuin reduktiojonossa $M \geq_\beta N$. □

Lemma 3.21. *Olkoon M ja N lambdatermejä. Jos on olemassa sellainen $\tilde{M} \in \tilde{\Lambda}$, jolla $|\tilde{M}| \overset{\alpha}{=} M$ ja $\varphi(\tilde{M}) \overset{\alpha}{=} N$, niin $M \geq_\beta N$ (ks. kuva 3.3).*

Todistus. Termit $\tilde{M}$ ja M ovat taas identtisiä lukuunottamatta mahdollisia aaltoviivoja. Termi N saadaan beetaredusoimalla termistä M ne lambdat, jotka termissä $\tilde{M}$ ovat aaltoviivattuja. □

Kuva 3.3: Lemma 3.21

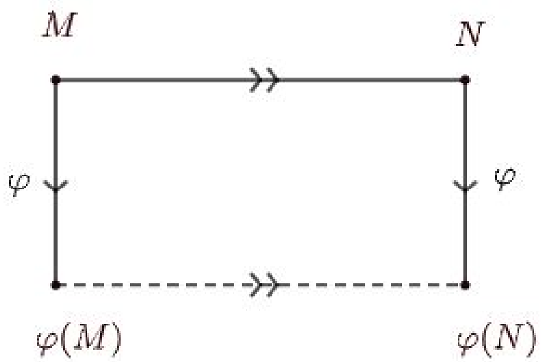


Lemma 3.22. *Olkoon M ja N laajennetun lambdalaskennan termejä. Seuraavat kohdat pätevät:*

i) $\varphi(M[x := N]) \stackrel{\alpha}{=} \varphi(M)[x := \varphi(N)]$.

ii) Jos $M \geq_{\tilde{\beta}} N$, *niin* $\varphi(M) \geq_{\beta} \varphi(N)$ *(ks. kuva 3.4).*

Kuva 3.4: Lemman 3.22 kohta ii)



Todistus. i) Väite todistetaan induktiolla termin M rakenteen suhteen. Koska tavallisille lambdatermeille funktio φ ei tee yhtään mitään, kohtaa 4 lukuunottamatta askeleet ovat käsitteellisesti yksinkertaisia.

1. Termi M on joko muuttuja x tai jokin muu muuttuja y. Ensimmäisessä tapauksessa
$$\varphi(x[x := N]) \stackrel{\alpha}{=} \varphi(N) \stackrel{\alpha}{=} \varphi(x)[x := \varphi(N)].$$

Toisessa tapauksessa

$$\varphi(y[x := N]) \overset{\alpha}{=} \varphi(y) \overset{\alpha}{=} \varphi(y)[x := \varphi(N)].$$

2. Oletetaan, että $M = (PQ)$ joillakin $P, Q \in \tilde{\Lambda}$, joilla väite i) pätee, ja P ei ole muotoa $(\tilde{\lambda}y.P')$. Tällöin

$$\begin{aligned}\varphi((PQ)[x := N]) &\overset{\alpha}{=} \varphi(P[x := N]Q[x := N]) \overset{\alpha}{=} \varphi(P[x := N])\varphi(Q[x := N])\\ &\overset{\alpha}{=} \varphi(P)[x := \varphi(N)]\varphi(Q)[x := \varphi(N)] \overset{\alpha}{=} \varphi(PQ)[x := \varphi(N)].\end{aligned}$$

3. Oletetaan, että $M = (\lambda y.P)$ jollakin $P \in \tilde{\Lambda}$, jolla väite i) pätee. Muuttujaoletuksen (3.8) perusteella sidottu muuttuja y ei esiinny vapaana termissä N ja $y \neq x$. Nyt

$$\begin{aligned}\varphi((\lambda y.P)[x := N]) &\overset{\alpha}{=} \varphi(\lambda y.(P[x := N])) \overset{\alpha}{=} \lambda y.\varphi(P[x := N])\\ &\overset{\alpha}{=} \lambda y.\varphi(P)[x := \varphi(N)] \overset{\alpha}{=} \varphi(\lambda y.P)[x := \varphi(N)].\end{aligned}$$

4. Oletetaan, että $M = ((\tilde{\lambda}y.P)Q)$ joillakin $P, Q \in \tilde{\Lambda}$, joilla väite i) pätee. Muuttujaoletuksen perusteella y ei esiinny vapaana termeissä N ja Q sekä $y \neq x$. Nyt

$$\begin{aligned}\varphi\big(((\tilde{\lambda}y.P)Q)[x := N]\big) &\overset{\alpha}{=} \varphi\big((\tilde{\lambda}y.P[x := N])(Q[x := N])\big)\\ &\overset{\alpha}{=} \varphi(P[x := N])\big[y := \varphi(Q[x := N])\big]\\ &\overset{\alpha}{=} \varphi(P)[x := \varphi(N)]\big[y := \varphi(Q)[x := \varphi(N)]\big].\end{aligned}$$

   Tarkastellaan yllä olevia sijoituksia. Muuttuja y ei ole vapaana termissä N, joten se ei myöskään ole vapaana termissä $\varphi(N)$. Tällöin sijoitus $[x := \varphi(N)]$ ei vaikuta y:n esiintymiin, joten sijoitusten $[x := \varphi(N)]$ ja $[y := ...]$ järjestyksen vaihtaminen keskenään ei vaikuta lopputulokseen ainakaan y:n sijoituksen kannalta.

   Toisaalta, jos katsomme sijoitusta $\big[y := \varphi(Q)[x := \varphi(N)]\big]$, huomaamme, että kaikkien uusien muuttujan x esiintymien paikalle sijoitetaan termi $\varphi(N)$. Ei ole merkitystä tehdäänkö tämä sijoitus y:n sijoituksen yhteydessä vai vasta sen jälkeen. Saamme yhtäsuuruudet

$$\begin{aligned}\varphi(P)[x := \varphi(N)]\big[y := \varphi(Q)[x := \varphi(N)]\big] &\overset{\alpha}{=} \varphi(P)[y := \varphi(Q)][x := \varphi(N)]\\ &\overset{\alpha}{=} \varphi((\tilde{\lambda}y.P)Q)[x := \varphi(N)].\end{aligned}$$

ii) Oletetaan, että termi M on redusoituva termiin N. Väite $\varphi(M) \geq_\beta \varphi(N)$ todistetaan induktiolla reduktiojonon $M \geq_{\tilde{\beta}} N$ pituuden n suhteen: redusoituvuuden nojalla on olemassa termit $M_0, ..., M_n$, joilla

$$M \overset{\alpha}{=} M_0 \overset{\beta}{\to} M_1 \overset{\beta}{\to} ... \overset{\beta}{\to} M_n \overset{\alpha}{=} N.$$

- Jos $n = 0$, niin $M \overset{\alpha}{=} N$ ja täten myös $\varphi(M) \overset{\alpha}{=} \varphi(N)$.
- Oletetaan, että $M_n \xrightarrow{\beta} M_{n+1} \overset{\alpha}{=} N$ jollain $n \geq 0$ ja $\varphi(M) \geq_\beta \varphi(M_n)$. Tavoite on näyttää, että $\varphi(M_n) \geq_\beta \varphi(M_{n+1})$, josta seuraa $\varphi(M) \geq_\beta \varphi(N)$.

  Beetareduktio $M_n \xrightarrow{\beta} M_{n+1}$ redusoi jonkin termin M_n alitermin, joka on muotoa $(\lambda x.P)Q$ tai $(\tilde{\lambda} x.P)Q$.
$$(\lambda x.P)Q \xrightarrow{\beta} P[x := Q].$$

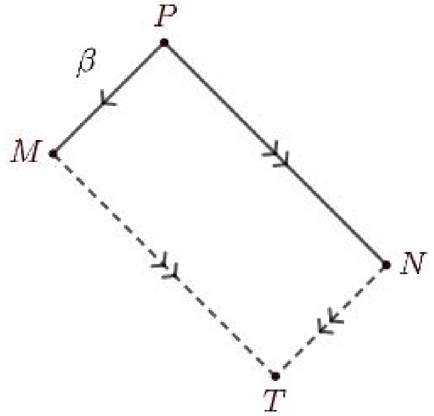
  Suoritetaan termin M_n alitermien suhteen induktio, jolla todistetaan, ettei ole väliä mille alitermille beetareduktio toteutuu. Periaatteessa ensimmäisen kohdan jälkeen loput seuraavat siitä, että beetareduktio – ja siten redusoituvuus – on määritelty nimenomaan alitermeille.

  1. Jos M_n on itse kyseinen alitermi $(\lambda x.P)Q$, niin M_{n+1} on termi $P[x := Q]$. Näytetään, että $\varphi(M_n) \geq_\beta \varphi(M_{n+1})$.
$$\varphi((\lambda x.P)Q) \overset{\alpha}{=} (\lambda x.\varphi(P))\varphi(Q) \xrightarrow{\beta} \varphi(P)[x := \varphi(Q)] \overset{\alpha}{=} \varphi(P[x := Q]).$$
     Viimeinen yhtäsuuruus seuraa kohdasta i). Todistus on täsmälleen sama alitermille $(\tilde{\lambda} x.P)Q$.
  2. Tässä ja seuraavissa kohdissa R ja S ovat M_n:n alitermejä. Oletetaan, että $M_n = (RS)$ ja $M_{n+1} = (RS')$ missä $S \xrightarrow{\beta} S'$ ja $\varphi(S) \geq_\beta \varphi(S')$. Tällöin
$$\varphi(RS) \overset{\alpha}{=} \varphi(R)\varphi(S) \geq_\beta \varphi(R)\varphi(S') \overset{\alpha}{=} \varphi(RS'),$$
     joten $\varphi(M_n) \geq_\beta \varphi(M_{n+1})$. Tulos osoitetaan samalla tavalla termien R ja S ollessa toisin päin.
  3. Oletetaan, että $M_n = (\lambda x.R)$ ja $M_{n+1} = (\lambda x.R')$, missä $R \xrightarrow{\beta} R'$ ja $\varphi(R) \geq_\beta \varphi(R')$. Tällöin
$$\varphi(\lambda x.R) \overset{\alpha}{=} \lambda x.\varphi(R) \geq_\beta \lambda x.\varphi(R') \overset{\alpha}{=} \varphi(\lambda x.R').$$
  4. Oletetaan, että $M_n = (\tilde{\lambda} x.R)S$ ja $M_{n+1} = (\tilde{\lambda} x.R')S$, missä $R \xrightarrow{\beta} R'$ ja $\varphi(R) \geq_\beta \varphi(R')$. Tällöin
$$\varphi((\tilde{\lambda} x.R)S) \overset{\alpha}{=} \varphi(R)[x := \varphi(S)] \geq_\beta \varphi(R')[x := \varphi(S)] \overset{\alpha}{=} \varphi((\tilde{\lambda} x.R')S).$$
     Termille S vastaava tulos osoitettiin kohdassa 2.

$\square$

Lemma 3.23 (Liuskalemma)**.** *Olkoon P, M ja N lambdatermejä. Jos $P \xrightarrow{\beta} M$ ja $P \geq_\beta N$, niin on olemassa termi T, johon sekä M että N ovat redusoituvia (ks. kuva 3.5).*

Kuva 3.5: Liuskalemma

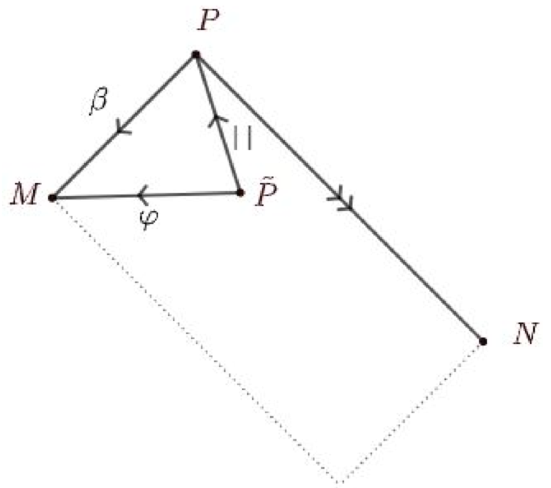


Todistus. Oletetaan, että $P \xrightarrow{\beta} M$ ja $P \geq_\beta N$ joillakin lambdatermeillä P, M ja N. Beetareduktiossa $P \xrightarrow{\beta} M$ redusoidaan yksi λ-abstraktio. Olkoon $\tilde{P} \in \tilde{\Lambda}$ muuten kuin termi P, mutta tämä kyseinen λ on korvattu merkillä $\tilde{\lambda}$. Nyt $\varphi(\tilde{P}) \overset{\alpha}{=} M$ ja $|\tilde{P}| \overset{\alpha}{=} P$ (ks. kuva 3.6).

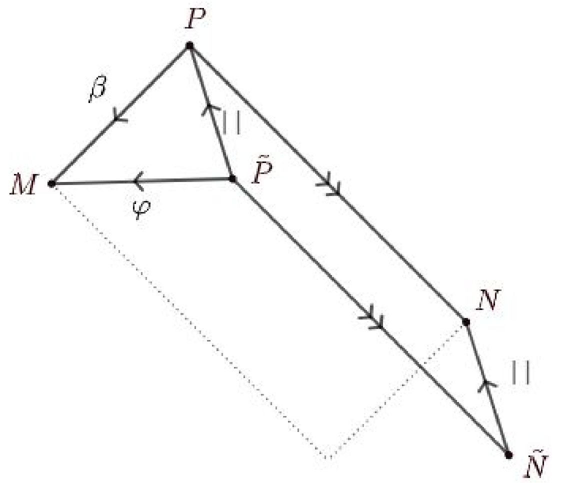
Käytetään lemmaa (3.20) eli muodostetaan termi $\tilde{N} \in \tilde{\Lambda}$ siten, että termistä $\tilde{P}$ redusoidaan kaikki vastaavat sovellukset kuin reduktiojonossa $P \geq_\beta N$. Nyt $|\tilde{N}| \overset{\alpha}{=} N$ ja $\tilde{P} \geq_{\tilde{\beta}} \tilde{N}$ (ks. kuva 3.7).

Haluttu termi T saadaan kun termistä $\tilde{N}$ redusoidaan korvamerkitty $\tilde{\lambda}$. Määritellään siis $T = \varphi(\tilde{N})$. Nyt T saadaan myös beetaredusoimalla tätä vastaava lambda termistä N. Toisin sanoen lemman (3.21) nojalla $N \geq_\beta T$ (ks. kuva 3.8).

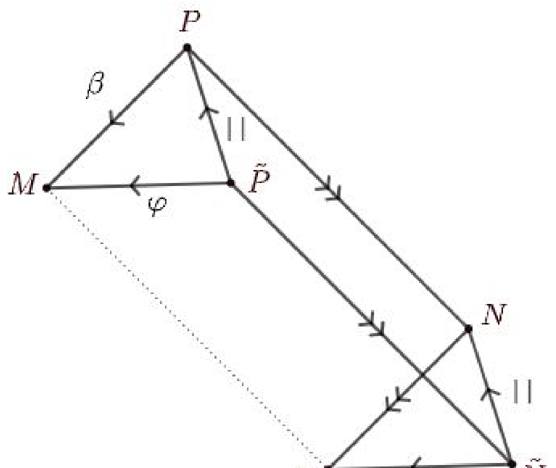
Toinen haluttu redusoituvuus $M \geq_\beta T$ saadaan lemman (3.22) kohdan ii) avulla: koska $\tilde{P} \geq_{\tilde{\beta}} \tilde{N}$, niin $\varphi(\tilde{P}) \geq_\beta \varphi(\tilde{N})$. □



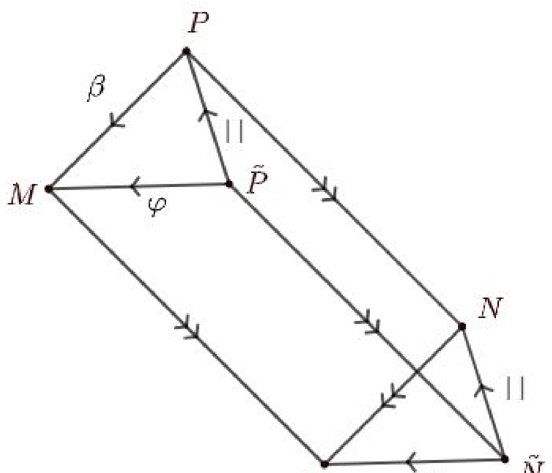
Kuva 3.6: Vaihe 1



Kuva 3.7: Vaihe 2 lemman 3.20 avulla



Kuva 3.8: Vaihe 3 lemman 3.21 avulla



Kuva 3.9: Vaihe 4 lemman 3.22 kohdan ii) avulla

Todistus. (Churchin-Rosserin lause) Oletetaan, että lambdatermi P on redusoituva lambdatermeihin M ja N. Suoritetaan induktio reduktiojonon $P \geq_\beta M$ pituuden n suhteen: jollakin $n \geq 0$ on olemassa termit $P_0, ..., P_n$, joilla

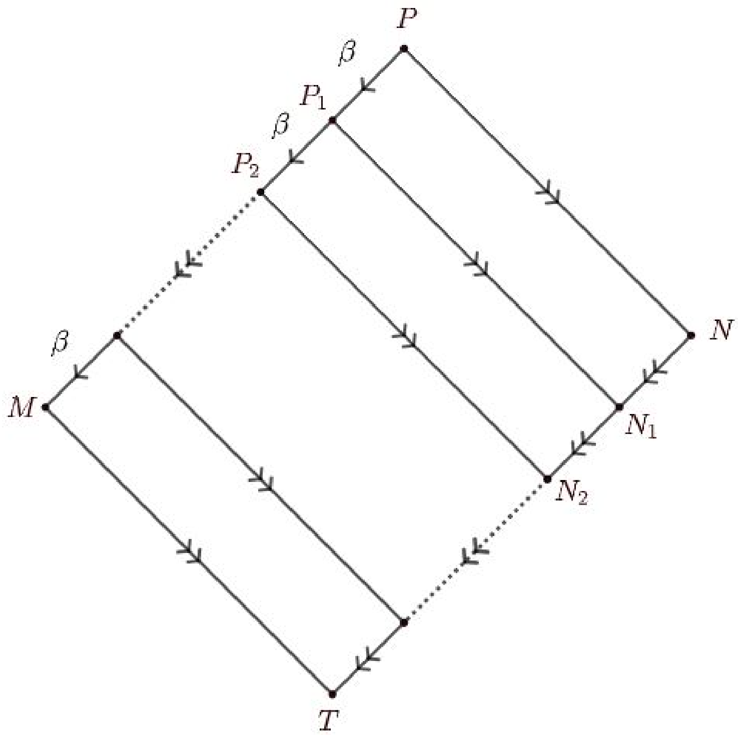
$$P \overset{\alpha}{\equiv} P_0 \overset{\beta}{\to} P_1 \overset{\beta}{\to} ... \overset{\beta}{\to} P_n \overset{\alpha}{\equiv} M.$$

Liuskalemmaa toistuvasti soveltamalla löydetään termi T (ks. kuva 3.10).

- Jos $n = 0$, niin $P \overset{\alpha}{\equiv} M$, ja voidaan määritellä $N_0 = N$ johon P_0 ja N ovat redusoituvia.
- Oletetaan, että $P_n \overset{\beta}{\to} P_{n+1} \overset{\alpha}{\equiv} M$ jollain $n \geq 0$, sekä on olemassa termi N_n, johon P_n ja N ovat redusoituvia. Liuskalemman nojalla on olemassa N_{n+1}, johon P_{n+1} ja N_n ovat redusoituvia. Termi N on redusoituva termiin N_n, joten se on redusoituva myös termiin N_{n+1}

Reduktiojonon pituuden ollessa n, termi N_n on sellainen, johon sekä N että M ovat redusoituvia. Määritellään siis haluttu $T = N_n$. □

Kuva 3.10: Churchin-Rosserin lause liuskalemman avulla



3.2 Kiintopistelause

Kurt Gödel todisti epätäydellisyyslauseensa itseensä viittaavalla logiikan kaavalla. Rekursiivisten funktioiden kiintopistelause mahdollistaa samankaltaisten itseään käsittelevien funktioiden löytämisen. Jotta luonnollisten lukujen funktio voisi käsitellä funktioita, tarvitaan koodaus funktioiden ja luonnollisten lukujen välillä, mikä tekee lauseen todistuksesta työlään. Lambdalaskennassa ei taas eritellä funktioiden ja niiden parametrien tyyppiä, jolloin vastaava tulos lambdatermeille on läpinäkyvämpi.

Funktion *kiintopiste* on objekti, joka kuvautuu funktion kautta itselleen. Esimerkiksi funktion $f(x) = x^3$ kiintopisteitä ovat 0 ja 1, koska $f(0) = 0$ ja $f(1) = 1$. Lambdalaskennan kiintopistelause sanoo, että jokaisella termillä on kiintopiste. Eli jokaiselle termille M on olemassa termi X, jolla

$$MX \overset{\beta}{=} X.$$

Vielä kiinnostavampaa on se, että tämän lisäksi on olemassa operaattori $\mathcal{Y}$, jolla kyseisen termin X voi löytää.

Lause 3.24 (Kiintopistelause)**.** *On olemassa* kiintopisteoperaattori $\mathcal{Y}$*, jolla jokaiselle lambdatermille M pätee:*

$$\mathcal{Y}M \overset{\beta}{=} M(\mathcal{Y}M).$$

Todistus. Olkoon M ensin kiinnitetty lambdatermi. Haluamme löytää termin X, joka tekee itsestään 'kopion', johon termiä M sovelletaan:

$$X \overset{\beta}{\to} MX.$$

Aikaisemmin määritelty $\Delta = \lambda x.xx$ toimii niin että termi $\Delta\Delta$ monistuu:

$$\Delta\Delta \overset{\beta}{\to} \Delta\Delta.$$

Haluttu X voidaan johtaa termiä Δ muokkaamalla. Termi Δ_M ei pelkästään monista lauseketta $\Delta_M\Delta_M$, vaan soveltaa termin M siihen, kun määritellään

$$\Delta_M = \lambda x.M(xx).$$

Nyt $\Delta_M\Delta_M \overset{\beta}{\to} M(\Delta_M\Delta_M)$. Kiintopiste X kiinnitetylle termille M on siis

$$\Delta_M\Delta_M = (\lambda x.M(xx))(\lambda x.M(xx)).$$

Haluamme kiintopisteoperaattorin $\mathcal{Y}$, jolla mielivaltaiselle M pätee:

$$\mathcal{Y}M \overset{\beta}{=} (\lambda x.M(xx))(\lambda x.M(xx)).$$

Tästä voimme abstrahoida M:n pois ja määritellä

$$\mathcal{Y} = \lambda y.(\lambda x.y(xx))(\lambda x.y(xx)).$$

$\square$

Todistuksessa saattoi panna merkille erään eriskummallisen seikan. Sen sijaan, että termin M kiintopistettä olisi haettu termistä M käsin, lähdettiin nimenomaan liikkeelle sen kiintopisteestä X, joka redusoituu termiin MX.

$$X \xrightarrow{\beta} MX.$$

Voi näyttää paradoksaalisesti siltä, että X jollain tavalla itse sisältäisi itsensä, mutta kuten nähtiin, termi $X = \mathcal{Y}M$ sisältää vain viittauksen itseensä toteutumattoman beetareduktion kautta. Mitä hyötyä on tällaisista kiintopisteistä? Pohditaan funktiota F, jolla pätee

$$F \stackrel{\beta}{=} MF.$$

Voisi sanoa, että F on määritelty sen itsensä ehdoilla jonkin apufunktion M avulla. Kyseessä on hyvin geneerinen rekursion muoto, jolla voi toteuttaa sekä primitiivirekursion että minimalisoinnin (ks. lauseet 5.2 ja 5.3).

Luku 4

Lambda-määriteltävät funktiot

Miten luonnollisten lukujen laskettavia funktioita voidaan esittää lambdalausekkeina? Tähän asti käyttämämme laskenta on ollut pelkkää syntaktista merkkijonomanipulaatiota ilman semantiikkaa. Kiinnostavaa on, ettei lambdalaskenta tarvitse enempää koneistoa luonnollisten lukujen määrittelemiseen, vaan ne ovat epäsuorasti jo osana rakentamaamme järjestelmää.

Alonzo Church keksi, että funktiosovellusten iteraatioiden lukumäärällä saadaan vastaavuus luonnollisten lukujen kanssa. Jos f on funktio ja x parametri, niin lukua n voi ajatella luonnehdittavaksi lambdatermillä $\overline{n}$, jossa funktiota f on n kertaa sovellettu termiin x. Saadaan

$$\begin{aligned}\overline{0} &:= \lambda fx.x \\ \overline{1} &:= \lambda fx.fx \\ \overline{2} &:= \lambda fx.f(fx) \\ &\vdots\end{aligned}$$

Näitä luonnollisten lukujen koodauksia kutsutaan nimellä *Churchin numeraalit.* Jokainen numeraali $\overline{n}$ on kaksiparametrinen funktio, joka soveltaa ensimmäistä parametria n kertaa toiseen parametriin.

Esimerkki 4.1. Muista, että *currying*-merkinnällä $\lambda fx.x = \lambda f.(\lambda x.x)$. Funktiolla $\Delta = \lambda x.xx$ ja termeillä M ja N saadaan:

- $\overline{0}NM = (\lambda f.(\lambda x.x))NM \xrightarrow{\beta} (\lambda x.x)M \xrightarrow{\beta} M.$
- $\overline{1}\Delta M = (\lambda f.(\lambda x.fx))\Delta M \xrightarrow{\beta} (\lambda x.\Delta x)M \xrightarrow{\beta} \Delta M \xrightarrow{\beta} MM.$
- $\overline{2}\Delta M \overset{\beta}{=} \Delta(\Delta M) \xrightarrow{\beta} \Delta(MM) \overset{\beta}{=} (MM)(MM).$

Käytetään numeraaleille seuraavaa induktiivista määritelmää.

Määritelmä 4.2. Churchin numeraalit.

$$\overline{0} = \lambda fx.x$$
$$\overline{n+1} = \lambda fx.f(\overline{n}fx)$$

Huomautus 4.3. Tässä kirjaimia f ja x on käytetty vain selvennyksen vuoksi: koska lambdatermeistä ei ole eroteltavissa muuttujia ja funktioita, niin yhtä hyvin voitaisiin määritellä $\overline{0} = \lambda ab.b$.

Huomautus 4.4. Kaikki Churchin numeraalit ovat normaalimuodossa. Lisäksi koska normaalimuodot ovat Churchin-Rosserin lauseen korollaarin (3.16) mukaan yksikäsitteisiä, niin minkään termin ei ole mahdollista sieventyä kahteen eri numeraaliin.

Kun luonnolliset luvut on määritelty lambdalaskennan termein, voidaan viimein puhua luonnollisten lukujen funktioista.

Määritelmä 4.5. Funktio $f\colon \mathbb{N}^n \to \mathbb{N}$ on *lambda-määriteltävä*, jos on olemassa lambdatermi F, jolla

$$f(x_1, ..., x_n) = b, \quad \text{jos ja vain jos} \quad F\overline{x_1}...\overline{x_n} \overset{\beta}{=} \overline{b}.$$

Tällöin sanotaan, että termi F määrittää funktion f.

Lause 4.6. *Seuraavat funktiot ovat lambda-määriteltäviä:*

i) Vakiofunktio nollalle $Z(a) = 0$

Todistus. Z voidaan määritellä termillä $\lambda a.\overline{0}$. □

ii) Projektiofunktio $P_i^n(a_1, ..., a_n) = a_i$, *missä* $1 \leq i \leq n$

Todistus. P_i^n voidaan määritellä termillä $\lambda a_1...a_n.a_i$. □

iii) Seuraajafunktio $S(a) = a + 1$

Todistus. Numeraali $\overline{n}$ edustaa jonkin funktion f n-kertaista iteraatiota parametriin x. Termi $\overline{n+1}$ on siis sellainen, jossa funktiota f iteroidaan yhden kerran enemmän. S voidaan määritellä termillä $\boldsymbol{S} = \lambda afx.f(afx)$. On tärkeää muistaa, että *currying*-merkinnän mukaan

$$\lambda afx.M = \lambda a.(\lambda fx.M).$$

Jos $n \in \mathbb{N}$, niin

$$\boldsymbol{S}\overline{n} = (\lambda a.(\lambda fx.f(afx)))\overline{n} \xrightarrow{\beta} \lambda fx.f(\overline{n}fx) = \overline{n+1}.$$

Huomaa, että funktion parametrin ollessa jotain muuta kuin Churchin numeraali, tulokseksi voi tulla jotain missä ei ole mieltä. Emme kuitenkaan ole kiinnostuneita tällaisista syötteistä. □

iv) $a + b$

Todistus. Iteraationa ajateltuna summa $a + b$ on seuraajafunktion S soveltamista b kertaa lukuun a. Churchin numeraali $\overline{b}$ on funktio, joka soveltaa jotain funktiota f johonkin parametriin x yhteensä b kertaa. Jos siis ajattelemme, että $f = \boldsymbol{S}$ ja $x = \overline{a}$, niin $\overline{a + b} = \overline{b}\boldsymbol{S}\overline{a}$. Siten summan määrittävä termi on

$$\lambda ab.b\boldsymbol{S}a.$$

□

Edeltäjäfunktio Churchin numeraaleille on seuraajaa hankalampi. Haluaisimme parametrille $\overline{n}$ palauttaa numeraalin $\overline{n \dot{-} 1}$. Keino tehdä tämä on muodostaa pareja $(\overline{0}, \overline{1}), (\overline{1}, \overline{2}), (\overline{2}, \overline{3}), ...$, kunnes parin toinen jäsen on $\overline{n}$, ja palauttaa sitten parin ensimmäinen jäsen.

Määritellään funktio $\boldsymbol{pari}$ joka ottaa parametrikseen lambdatermit M ja N sekä simuloi järjestettyä paria (M, N). Lisäksi tarvitaan funktiot $\boldsymbol{eka}$ ja $\boldsymbol{toka}$, jotka antavat parin jäsenet. Menetelmä muistuttaa tietokoneohjelmointia. Numeraalien ohella tämä koodaus on Churchin idea, ja kiehtova esimerkki datan (pari) ja prosessin (funktio) eron häilyvyydestä.

$$\boldsymbol{pari} = \lambda xyz.zxy.$$

Kun kyseiselle funktiolle antaa parametrit M ja N, saadaan funktio, joka ottaa funktioparametrin, jota sovelletaan termeihin M ja N:

$$(\boldsymbol{pari}\ M\ N) \overset{\beta}{=} \lambda z.zMN.$$

Kun tälle funktiolle antaa parametriksi projektiofunktion $(\lambda xy.x)$, saadaan vastaukseksi M. Parametrilla $(\lambda xy.y)$ saadaan taas vastaus N. Funktiota $\boldsymbol{pari}$ on tarkoitus soveltaa funktioihin $\boldsymbol{eka}$ ja $\boldsymbol{toka}$, jotka hakevat parin jäsenet. Määritellään

$$\begin{aligned} \boldsymbol{eka} &= \lambda p.p(\lambda xy.x), \text{ ja} \\ \boldsymbol{toka} &= \lambda p.p(\lambda xy.y). \end{aligned}$$

Nyt

$$\begin{aligned} (\boldsymbol{eka}\ (\boldsymbol{pari}\ M\ N)) &\overset{\beta}{=} M, \text{ ja} \\ (\boldsymbol{toka}\ (\boldsymbol{pari}\ M\ N)) &\overset{\beta}{=} N. \end{aligned}$$

Tämän jälkeen on suoraviivaista rakentaa edeltäjäfunktio. Merkitään seuraajafunktiota kirjaimella $\boldsymbol{S}$. Funktio $\boldsymbol{S^2}$ on sellainen, joka parametrin ollessa pari $(\overline{n}, \overline{m})$, sivuuttaa ensimmäisen jäsenen ja palauttaa parin $(\overline{m}, \overline{m+1})$.

$$\boldsymbol{S^2} = \lambda p.\big(\boldsymbol{pari}\ (\boldsymbol{toka}\ p)\ (\boldsymbol{S}\ (\boldsymbol{toka}\ p))\big).$$

Kun lähdetään parista $(\overline{0}, \overline{0})$, niin soveltamalla n kertaa funktiota $\boldsymbol{S^2}$ saadaan pari $(\overline{n \dot{-} 1}, \overline{n})$. Numeraalin $\overline{n}$ edeltäjä tietysti saadaan tästä parista ensimmäisenä. Edeltäjäfunktion määrittelevä termi on

$$\boldsymbol{P} = \lambda n.\big(\boldsymbol{eka}\ (n\ \boldsymbol{S^2}\ (\boldsymbol{pari}\ \overline{0}\ \overline{0}))\big).$$

Tällä konstruktiolla olemme todistaneet seuraavan lauseen.

Lause 4.7. *Seuraavat funktiot ovat lambda-määriteltäviä:*

i) Edeltäjäfunktio $a \dot{-} 1 = \begin{cases} 0, & \text{jos } a = 0 \\ a - 1, & \text{muuten} \end{cases}$

ii) $a \dot{-} b = \begin{cases} 0, & \text{jos } a \leq b \\ a - b, & \text{muuten} \end{cases}$

Todistus. Erotus määritellään termillä $\lambda ab.b\boldsymbol{P}a$. □

Määriteltävien funktioiden kirjo laajenee entisestään funktioiden yhdisteillä.

Lause 4.8. *Lambda-määriteltävien funktioiden luokka on suljettu yhdistämisen suhteen: jos* $f\colon \mathbb{N}^n \to \mathbb{N}$ *ja* $g_i\colon \mathbb{N}^m \to \mathbb{N}$ *ovat lambda-määriteltäviä, kun* $1 \leq i \leq n$*, niin yhdistetty funktio*

$$h(x_1, ..., x_m) = f(g_1(x_1, ..., x_m), ..., g_n(x_1, ..., x_m))$$

on lambda-määriteltävä.

Todistus. Olkoon F ja G_i funktiot f ja g_i määrittävät termit. Funktio h on mahdollista määritellä termillä

$$\lambda x_1 ... x_m . F(G_1 x_1 ... x_m) ... (G_n x_1 ... x_m).$$

□

Tämän jälkeen emme välttämättä tarvitse täsmällistä (ja usein monimutkaista) lambdatermiä osoittaessamme lambda-määriteltävyyttä.

Esimerkki 4.9. Funktio $|x - y|$ on lambda-määriteltävä, sillä $|x - y| = (x \dot{-} y) + (y \dot{-} x)$.

Seuraava lause vielä näyttää, että lambda-määriteltävyys on suljettu hyvin tavallisen muodon suhteen.

Lause 4.10. *Jos* $f_1 \colon \mathbb{N}^n \to \mathbb{N}$ *ja* $f_2 \colon \mathbb{N}^n \to \mathbb{N}$ *ovat lambda-määriteltäviä sekä* $\boldsymbol{x} \in \mathbb{N}^n$*, niin paloittain määritelty funktio*

$$h(y, \boldsymbol{x}) = \begin{cases} f_1(\boldsymbol{x}), & \textit{jos} \quad y = 0 \\ f_2(\boldsymbol{x}), & \textit{muuten} \end{cases}$$

on lambda-määriteltävä.

Lauseen todistuksessa käytettävä menetelmä on melkeinpä oleellisempi, kuin tulos itse. Tahdotaan, että lambdatermin laskenta haarautuu kahteen eri polkuun jonkinlaisen totuustestauksen perusteella. Koodataan totuusarvot *tosi* ja *epätosi* projektiofunktioiksi: merkitään $\boldsymbol{T} = \lambda xy.x$ ja $\boldsymbol{E} = \lambda xy.y$. Nyt jos P on totuusarvo (eli joko $P = \boldsymbol{T}$ tai $P = \boldsymbol{E}$), niin ehtolause

$$\text{Jos } P, \text{ niin } M, \text{ muuten } N$$

voidaan esittää lambdalausekkeena

$$PMN.$$

Todistus. (Lause (4.10)) Jotta h saadaan määriteltyä, tarvitaan nollatestausfunktio ε, joka numeraalille y palauttaa totuusarvon $\boldsymbol{T}$, jos $y \stackrel{\beta}{=} \overline{0}$, ja muussa tapauksessa $\boldsymbol{E}$. Muista, että numeraali $\overline{0}$ on funktio, joka kahdelle parametrille f ja x palauttaa funktion f sovelluksen parametriin x nolla kertaa: $\overline{0}fx \stackrel{\beta}{=} x$. Nollatesti ε määritellään

$$\varepsilon = \lambda z.z(\lambda x.\boldsymbol{E})\boldsymbol{T}.$$

Jos parametri $z \stackrel{\beta}{=} \overline{0}$, niin ylläoleva palauttaa $\boldsymbol{T}$. Kaikilla muilla Churchin numeraaleilla funktiota $(\lambda x.\boldsymbol{E})$ sovelletaan termiin $\boldsymbol{T}$ vähintään kerran. Olkoon F_1 ja F_2 funktiot f_1 ja f_2 määrittävät termit, sekä merkitään vastaavia lambdamuuttujia myös $\boldsymbol{x} = x_1 ... x_n$. Nyt paloittain määrittely

$$\text{Jos } (y = 0), \text{ niin } f_1(\boldsymbol{x}), \text{ muuten } f_2(\boldsymbol{x})$$

ilmaistaan terminä

$$(\varepsilon y)(F_1\boldsymbol{x})(F_2\boldsymbol{x}).$$

Siis funktion h määrittelevä termi on

$$\lambda y\boldsymbol{x}.(\varepsilon y)(F_1\boldsymbol{x})(F_2\boldsymbol{x}).$$

□

Huomautus 4.11. Jos-termejä ketjuttamalla tulos on mahdollista yleistää mielivaltaiseen määrään tapauksia:

$$\text{Jos } (y = 0), \text{ niin } f_1, \text{ muuten } \big(\text{Jos } (y = 1), \text{ niin } f_2, \text{ muuten } f_3\big).$$

Tämän tutkielman tuloksiin kuitenkin riittää kahden palasen tapaukset, joten yleisemmän muodon todistus sivuutetaan.

Luku 5

Yhtenevyys

Vuonna 1931 Kurt Gödel oli todistanut epätäydellisyyslauseensa koneellistamalla predikaattilogiikan rekursiivisten funktioiden avulla. Tästä inspiroituneena Alonzo Churchin kanssa samanaikaisesti *Entscheidungsproblem*:ia ratkoi Alan Turing, joka kehitti Turingin koneet omaksi algoritmilaskennan määritelmäksi. Toistensa aikeista tietämättä Church ja Turing olivat keksineet täysin ekvivalentit laskennan mallit, kuten Turing myöhemmin osoitti. Tämän jälkeen on esitetty useita erilaisia koneellisen laskennan formalisointeja, jotka kaikki on näytetty yhteneviksi.

Laskettavuuden käsite näyttäisi siis olevan riippumaton mistään sitä esittävästä formaalista määritelmästä. Tämän johdosta laskettavuuden teorian työhypoteesina pidetään Churchin teesiä. Teesin mukaan rekursiiviset funktiot ovat täsmälleen ne funktiot, joiden arvot saavutetaan jollain (intuitiivisesti) mekaanisella prosessilla. Väitettä ei voi matemaattisesti todistaa, mutta laskennan mallien ekvivalenssi toimii vahvana empiirisenä näyttönä.

5.1 Rekursiivisten funktioiden esitys lambdatermein

Tämän luvun päämääränä on todistaa peruslause:

Lause 5.1. *Funktiolle f pätee:*

$$f \text{ on rekursiivinen, jos ja vain jos } f \text{ on lambda-määriteltävä.}$$

Ensin läpikäydään suunta vasemmalta oikealle: jokainen rekursiivinen funktio on lambda-määriteltävä. Todistus tehdään induktiolla rekursiivisten funktioiden määritelmän suhteen. Luvun 4 lauseissa (4.6) ja (4.8) on todistettu, että

1. nollafunktio, seuraajafunktio ja projektiofunktio ovat lambda-määriteltäviä ja

2. lambda-määriteltävien funktioiden luokka on suljettu yhdistämisen suhteen.

Induktiivinen todistus tarvitsee vielä seuraavat palaset:

3. lambda-määriteltävien funktioiden luokka on suljettu primitiivirekursion ja
4. minimalisoinnin suhteen.

Kappaleessa 3.2 esiteltiin kiintopisteoperaattori $\mathcal{Y}$, joka tietyssä mielessä on itseenviittauksen ruumiillistuma lambdalaskennassa. Operaattorin avulla voidaan löytää itsensä rekursiivisesti määritteleviä funktioita.

Lause 5.2. *Lambda-määriteltävien funktioiden luokka on suljettu primitiivirekursion suhteen: jos $f\colon \mathbb{N}^n \to \mathbb{N}$ ja $g\colon \mathbb{N}^{n+2} \to \mathbb{N}$ ovat lambda-määriteltäviä, niin primitiivirekursiolla saatu funktio*

$$\begin{cases} h(\boldsymbol{x}, 0) = f(\boldsymbol{x}) \\ h(\boldsymbol{x}, y+1) = g(\boldsymbol{x}, y, h(\boldsymbol{x}, y)) \end{cases}$$

on lambda-määriteltävä.

Tehdään selkeyden vuoksi ensin todistus tapaukselle $n = 0$, jolloin parametreista $\boldsymbol{x}$ ei tarvitse huolehtia. Tämän jälkeen yleinen todistus on mutkaton.

Todistus. ($n = 0$.) Nyt h on muotoa

$$\begin{cases} h(0) = a \\ h(y+1) = g(y, h(y)), \end{cases}$$

missä g on lamda-määritelty termillä G. Tavoite on löytää termi H, jolla Hy vastaa paloittain määrittelyä

$$\text{Jos } (y = 0), \text{ niin } a, \text{ muuten } g(y \dot{-} 1, h(y \dot{-} 1)).$$

Edeltäjäfunktio $y \dot{-} 1$ on lauseen (4.7) nojalla lambda-määriteltävä termillä $\boldsymbol{P}$. Olkoon nollatestifunktio ε kuten lauseen (4.10) todistuksessa. Halutaan, että Hy toteuttaa

$$Hy \overset{\beta}{=} (\varepsilon y)\, \overline{a}\, \big(G(\boldsymbol{P}y)(H(\boldsymbol{P}y))\big).$$

Saattaa nousta kysymys, voidaanko H määritellä itsensä ehdoilla. Näytetään kiintopisteoperaattorin avulla, että tämä tosiaan on mahdollista. Abstrahoidaan ensin y:

$$H \overset{\beta}{=} \lambda y.(\varepsilon y)\overline{a}\big(G(\boldsymbol{P}y)(H(\boldsymbol{P}y))\big),$$

ja sitten itse H: määritellään

$$\delta = \lambda hy.(\varepsilon y)\overline{a}\big(G(\boldsymbol{P}y)(h(\boldsymbol{P}y))\big).$$

Nyt haluttu funktio on funktion δ kiintopiste. Määritellään siis $H = \mathcal{Y}\delta$. □

Todistus. *(Yleinen tapaus.)* Nyt

$$\begin{cases} h(\boldsymbol{x},0) = f(\boldsymbol{x}) \\ h(\boldsymbol{x},y+1) = g(\boldsymbol{x},y,h(\boldsymbol{x},y)). \end{cases}$$

Merkitään vastaavia lambdamuuttujia myös $\boldsymbol{x} = x_1...x_n$. Jos F ja G ovat funktioiden f ja g määrittävät termit, niin haluttu H on sellainen, että

$$H\boldsymbol{x}y \overset{\beta}{=} (\varepsilon y)(F\boldsymbol{x})\big(G\boldsymbol{x}(\boldsymbol{P}y)(H\boldsymbol{x}(\boldsymbol{P}y))\big).$$

Siis termi H on funktion

$$\lambda h\boldsymbol{x}y.(\varepsilon y)(F\boldsymbol{x})\big(G\boldsymbol{x}(\boldsymbol{P}y)(h\boldsymbol{x}(\boldsymbol{P}y))\big)$$

kiintopiste. □

Lause 5.3. *Lambda-määriteltävien funktioiden luokka on suljettu minimalisoinnin suhteen: jos $f\colon \mathbb{N}^{n+1} \to \mathbb{N}$ on lambda-määriteltävä ja kaikilla $\boldsymbol{x} \in \mathbb{N}^n$ on olemassa $y \in \mathbb{N}$, jolla $f(\boldsymbol{x},y) = 0$, niin*

$$h(\boldsymbol{x}) = \mu y(f(\boldsymbol{x},y) = 0)$$

on lambda-määriteltävä.

Todistus. Rekursiivisesti voidaan määritellä $h(\boldsymbol{x}) = g(\boldsymbol{x},0)$, missä

$$g(\boldsymbol{x},y) = \begin{cases} y, & \text{jos} \quad f(\boldsymbol{x},y) = 0 \\ g(\boldsymbol{x},y+1), & \text{muuten.} \end{cases}$$

Kuten edellisen lauseen todistuksessa, halutaan G vastaamaan ehtolausetta

$$\text{Jos } (f(\boldsymbol{x},y) = 0), \text{ niin } y, \text{ muuten } g(\boldsymbol{x},y+1).$$

Olkoon F funktion f määrittävä termi. Merkitään seuraajafunktion $x+1$ lambdatermiä $\boldsymbol{S}$ ja etsitään G jolla

$$G\boldsymbol{x}y \overset{\beta}{=} \big(\varepsilon(F\boldsymbol{x}y)\big)y\big(G\boldsymbol{x}(\boldsymbol{S}y)\big).$$

G on siten termin

$$\lambda g\boldsymbol{x}y.\big(\varepsilon(F\boldsymbol{x}y)\big)y\big(g\boldsymbol{x}(\boldsymbol{S}y)\big)$$

kiintopiste, ja funktion h määrittävä termi

$$H = \lambda\boldsymbol{x}.G\boldsymbol{x}\overline{0}.$$

□

Lauseista (4.6), (4.8), (5.2) ja (5.3) saadaan tarvittavat palaset seuraavaan tulokseen.

Lause 5.4. *Jos funktio f on rekursiivinen, se on lambda-määriteltävä.* □

5.2 Lambdalaskennan rekursiivisuus

Lause 5.5. *Jos funktio f on lambda-määriteltävä, se on rekursiivinen.*

Jotta lambdatermeistä voidaan puhua rekursiivisten funktioiden avulla, pitää ne ensin koodata luonnollisiksi luvuiksi. Käytettävää menetelmää kutsutaan Gödel-luetteloinniksi (engl. *Gödel numbering*). Annetaan ensin jokaiselle yksittäiselle symbolille w yksikäsitteinen luku $\#(w)$:

w	λ	.	(	)	x_1	x_2	...
$\#(w)$	1	2	3	4	5	6	...

Itse termien yksilölliseen luettelontiin käytetään hyödyksi alkulukuhajotelmaa. Merkkijono, joka muodostuu merkeistä $w_1...w_n$ annamme järjestysluvun – tai *Gödel-luvun*

$$2^{\#(w_1)} \cdot 3^{\#(w_2)} \cdots p_n^{\#(w_n)},$$

missä p_n on n :nes alkuluku. Esimerkiksi termin $(\lambda x_1.x_1)$ Gödel-luku on

$$2^3 \cdot 3^1 \cdot 5^5 \cdot 7^2 \cdot 11^5 \cdot 13^4.$$

Tässä kappaleessa ei käytetä lambdalaskennan kirjoituskonventioita, vaan termit rakennetaan määritelmän mukaan. Toisin sanoen sovellukset (MN) ja abstraktiot $(\lambda x.M)$ kirjoitetaan aina sulkeiden kanssa eikä *currying*-merkintää käytetä.

Gödel-luetteloinnilla voimme käsitellä lambdatermejä luonnollisten lukujen rekursiivisilla funktioilla. Lauseessa (2.14) on todistettu, että jaollisuus $'x|y'$ on rekursiivinen relaatio sekä x:nnes alkuluku p_x on saatavissa rekursiivisella funktiolla. Määritellään loput tarvittavat rekursiiviset funktiot ja relaatiot. Seuraavissa kohdissa 'merkki x' tarkoittaa merkkiä w, jonka luku $\#(w)$ on x, ja 'merkkijono x' tarkoittaa merkkijonoa, jonka Gödelluku x on.

1. $(x)_y = \mu z\big[x = 0 \text{ tai } y = 0 \text{ tai } p_y^{z+1} \nmid x\big]$.
   $(x)_y$ on y:nnes merkki merkkijonossa x. Jos $x = 0$ tai $y = 0$, niin funktio antaa 'pienimmän luvun z', sanalla sanoen luvun 0.

2. $len(x) = \mu y\big[x = 0 \text{ tai } 2 \nmid x \text{ tai } \big((x)_y > 0 \text{ ja } (x)_{y+1} = 0\big)\big]$.
   $len(x)$ on merkkijonon x pituus. Jos x ei ole jaollinen luvulla $p_1 = 2$ – toisin sanoen sen ensimmäistä merkkiä ei ole määritelty – se määritellään tyhjäksi merkkijonoksi.

3. $R(x) = 2^x$.
   $R(x)$ on merkkijono, joka koostuu ainoastaan merkistä x. Esimerkiksi luvut $R(1)$ ja $R(2)$ edustavat merkkijonoja 'λ' ja '.'.

4. $x * y = \mu z\big[z > 0$ ja $(\forall n \leq len(x))[(n > 0) \rightarrow (x)_n = (z)_n]$ ja
$(\forall n \leq len(y))[(n > 0) \rightarrow (y)_n = (z)_{len(x)+n}]\big]$.
$x * y$ on merkkijonojen x ja y liitos peräkkäin.

5. $Su(x, i, y) = \mu z\big[(\exists u, v \leq x)\{x = u * R((x)_i) * v$ ja $z = u * y * v$ ja $i = len(u) + 1\}\big]$.
$Su(x, i, y)$ saadaan merkkijonosta x, kun i:nnes merkki korvataan merkkijonolla y.

6. Seuraava funktio antaa Gödel-luvun jokaiselle Churchin numeraalille. Termi $\overline{0}$ on määritelty $(\lambda x_1.(\lambda x_2.x_2))$. Rekursioaskel saadaan siitä, että oikeanpuoleinen x_2 korvataan aina merkkijonolla (x_1x_2):

$$\begin{aligned}&\overline{1} = (\lambda x_1.(\lambda x_2.(x_1x_2)))\\ &\overline{2} = (\lambda x_1.(\lambda x_2.(x_1(x_1x_2))))\\ &\vdots\end{aligned}$$

Numeraalissa $\overline{n}$ tämä korvattavan merkin x_2 paikkaluku on $(9 + 2n)$.

$$\begin{cases} N(0) = 2^3 \cdot 3^1 \cdot 5^5 \cdot 7^2 \cdot 11^3 \cdot 13^1 \cdot 17^6 \cdot 19^2 \cdot 23^6 \cdot 29^4 \cdot 31^4 \\ N(n+1) = Su(\ N(n),\ 9 + 2n,\ 2^3 \cdot 3^5 \cdot 5^6 \cdot 7^4\ ). \end{cases}$$

7. $Num(x) \equiv \exists n \leq x[x = N(n)]$.
$Num(x)$ pätee, jos ja vain jos merkkijono x on jokin Churchin numeraali $\overline{n}$ (tosin täsmälleen muuttujilla x_1 ja x_2 määriteltynä kuten edellisessä kohdassa).

8. $N^{-1}(x) = \mu n\big[\neg Num(x)$ tai $x = N(n)\big]$.
Jos x on Churchin numeraali $\overline{n}$, niin $N^{-1}(x) = n$. Jos x ei ole numeraali, funktion totaalisuuden vuoksi se antaa 'leikkiarvon' 0. Tätä funktiota tullaan käyttämään kuitenkin vain sellaisessa tilanteessa, jossa on jo valmiiksi tiedossa päteekö $Num(x)$.

Loput määritelmät tähtäävät relaatioiden $'M \xrightarrow{\beta} N'$ ja $'M \xrightarrow{\alpha} N'$ rekursiivisuuden todistamiseen. Ensin täytyy määritellä hyvin muodostetut lambdatermit.

9. $E(x) = R(3) * x * R(4)$.
$E(x)$ on merkkijono, joka saadaan kun merkkijonon x ympärille lisätään sulut.

10. $V(x) \equiv x \geq 5$.
Merkki x on muuttuja.

11. $Occ(v, x) \equiv V(v)$ ja $(\exists n \leq len(x))[v = (x)_n]$.
Muuttuja v esiintyy merkkijonossa x.

12. Tässä x ei edusta merkkiä tai merkkijonoa vaan jonoa merkkijonoista. Siis $x = 2^{h_1}3^{h_2}\cdots p_n^{h_n}$, missä luvut h_i ovat merkkijonoja.

$$\begin{aligned}\lambda(x) \equiv (\forall n \le len(x))\Big[&(0 < n) \rightarrow \\ &\Big\{(\exists v < x)\big[V(v) \text{ ja } (x)_n = R(v)\big] \text{ tai} \\ &(\exists p, q < n)\big[(0 < p, q) \text{ ja } \big\{(x)_n = E((x)_p * (x)_q) \text{ tai} \\ &\quad \big((x)_n = E(R(1) * (x)_p * R(2) * (x)_q) \text{ ja} \\ &\quad (\exists v \le x)[V(v) \text{ ja } (x)_p = R(v)]\big)\big\}\big]\Big\}\Big] \text{ ja } len(x) > 0.\end{aligned}$$

Jono x on alitermijono, eli x koostuu merkkijonoista, joista jokainen on joko muuttuja tai on koottu joistakin jonon aikaisemmista jäsenistä m ja n käyttäen rakenteita '(mn)' ja '$(\lambda x_i.m)$'.

13. $\Lambda(x) \equiv (\exists n \le p_{len^2(x)}^{x \cdot len^2(x)})\big[\lambda(n) \text{ ja } x = (n)_{len(n)}\big]$.

Merkkijono x on lambdatermi. Ylärajan $p_{len^2(x)}^{x \cdot len^2(x)}$ voi perustella seuraavasti: selvitetään lyhyimmän merkkijonon x määrittävän alitermijonon n pituuden jokin yläraja l. Tällöin kaikki n:n alkulukutekijät (x:n alitermit) ovat pienempiä tai yhtäsuuria kuin p_l, niiden lukumäärä on pienempää tai yhtäsuurta kuin l ja niiden potenssi on pienempää tai yhtäsuurta kuin x. Täten $n \le p_l^{x \cdot l}$.

Jonon n pituus on korkeintaan x:n alitermien lukumäärä, joka on taas korkeintaan x:n osamerkkijonojen lukumäärä. Osamerkkijonojen, joiden pituus on 1, lukumäärä on korkeintaan $len(x)$ ja osamerkkijonojen, joiden pituus on 2, lukumäärä on korkeintaan $len(x) - 1$. Kaikkien osamerkkijonojen lukumäärä on korkeintaan siis $len(x) + (len(x) - 1) + ... + 1 \le len^2(x)$.

14. $$\begin{aligned}Sid(v, n, x) \equiv V(v) \text{ ja } \Lambda(x) \text{ ja } (\exists a, b, c \le x)\big[&\\ &x = a * E(R(1) * R(v) * R(2) * b) * c \text{ ja } \Lambda(b) \text{ ja} \\ &len(a) < n \le len(a) + len(b) + 5\big].\end{aligned}$$

Muuttuja v on sidottu n:nen merkin kohdalla lambdatermissä x.

15. $Sid(v, x) \equiv (\exists n \le len(x))\big[v = (x)_n \text{ ja } Sid(v, n, x)\big]$.
Muuttuja v esiintyy sidottuna merkkijonossa x.

16. $Vap(v, n, x) \equiv V(v) \text{ ja } \Lambda(x) \text{ ja } v = (x)_n \text{ ja } n \le len(x) \text{ ja } \neg Sid(v, n, x)$.
Muuttuja v on vapaa n:nen merkin kohdalla lambdatermissä x.

17. $Vap(v, x) \equiv (\exists n \le len(x))[Vap(v, n, x)]$.
Muuttuja v esiintyy vapaana lambdatermissä x.

18. $S(x,v,y) = SR(len(x), x, v, y)$, missä

$$\begin{cases} SR(0,x,v,y) = x \\ SR(k+1,x,v,y) = \mu z\big[\big((x)_{k+1} \neq v \text{ ja } z = SR(k,x,v,y)\big) \text{ tai} \\ \qquad\qquad\qquad\qquad \big((x)_{k+1} = v \text{ ja } z = Su(\ SR(k,x,v,y),\ k+1,\ y\ )\big)\big]. \end{cases}$$

$S(x,v,y)$ on merkkijono x, missä kaikkien muuttujan v esiintymiin on sijoitettu merkkijono y.

19. $'x \overset{\alpha}{\rightarrow} y' \equiv \Lambda(x)$ ja $(\exists p,q,s,t \leq x)(\exists r \leq y)\Big[$
$$x = p * E(R(1) * R(q) * R(2) * s) * t \text{ ja}$$
$$V(q) \text{ ja } \Lambda(s) \text{ ja } V(r) \text{ ja } \neg Occ(r,s) \text{ ja}$$
$$y = p * E(R(1) * R(r) * R(2) * S(s,q,R(r))) * t\Big].$$

20. $'x \overset{\beta}{\rightarrow} y' \equiv \Lambda(x)$ ja $(\exists p,q,r,s,t \leq x)\Big[$
$$x = p * E\big(E(R(1) * R(q) * R(2) * r) * s\big) * t \text{ ja}$$
$$V(q) \text{ ja } \Lambda(r) \text{ ja } \Lambda(s) \text{ ja } \neg Sid(q,r) \text{ ja}$$
$$(\forall u \leq s)\{Vap(u,s) \rightarrow \neg Sid(u,r)\} \text{ ja}$$
$$y = p * S(r,q,s) * t\Big].$$

Määritelmän (3.11) mukainen redusoituvuus (ja siten beetayhtäsuuruus) mielivaltaisten termien välillä on osoitettu ratkeamattomaksi Turingin koneiden pysähtymisongelman tavoin. Relaatio $'x \overset{\beta}{=} y'$ ei siis ole rekursiivinen. Keskenään beetayhtäsuurten termien joukko on kuitenkin *rekursiivisesti luetteloituva* (engl. *recursively enumerable*). Tämä tarkoittaa sitä, että voimme esittää rekursiivisen funktion $\theta(x,y)$ joka luetteloi kaikki termin x kanssa beetayhtäsuuret termit:

$$\{\theta(x,0), \theta(x,1), \theta(x,2), ...\} = \{z | 'z \overset{\beta}{=} x'\}.$$

Tästä luetteloinnista voimme minimalisoinnilla hakea tarvitsemamme, jos vain voimme olla varmoja, että se tosiaan ennemmin tai myöhemmin tulee vastaan.

Luetteloiva funktio θ termille x saadaan seuraavalla menetelmällä. Tässä ω on Gödelluku lambdatermien jonolle, jota rakennamme askel kerrallaan. Lähdemme jonosta $\omega_0 = 2^x$. Ensimmäinen askel on jono $\omega_1 = 2^x \cdot 3^{x_1}$, missä x_1 on yhden beetareduktion tai alfamuunnoksen päässä termistä x. Toinen askel on jono $\omega_2 = 2^x \cdot 3^{x_1} \cdot 5^{x_2}$, missä x_2 on yhden beetareduktion tai alfamuunnoksen päässä joko termistä x tai termistä x_1. Kun valitsemme termit x_i kattavasti ja varmistamme, etteivät mitkään niistä ole samoja (yhtäsuuria), ennen pitkää jokainen beetayhtäsuuri termi on tässä listassa.

Luetteloinnin y:nnes jäsen $\theta(x,y)$ saadaan y:n askeleen jälkeen tehdystä jonosta viimeisenä.

21. $'x \overset{1}{=} y' \equiv \big[x \overset{\beta}{\to} y$ tai $x \overset{\alpha}{\to} y$ tai $y \overset{\beta}{\to} x$ tai $y \overset{\alpha}{\to} x\big]$.
Toinen lambdatermeistä x ja y on johdettavissa toisesta käyttämällä yhtä beetareduktiota tai alfamuunnosta.

22. $\theta(x,y) = \big(\omega(x,y)\big)_{y+1}$, missä

$$\begin{cases} \omega(x,0) = 2^x \\ \omega(x,k+1) = \omega(x,k) \cdot p_{k+2}\hat{\ }\mu z\Big[\\ \qquad\qquad (\forall n \leq len(\omega(x,k)))\big[\big(\omega(x,k)\big)_n \neq z\big] \text{ ja} \\ \qquad\qquad (\exists n \leq len(\omega(x,k)))\big[\big(\omega(x,k)\big)_n \overset{1}{=} z\big]\Big]. \end{cases}$$

Todistus. Lause (5.5) Olkoon funktio $f\colon \mathbb{N}^n \to \mathbb{N}$ lambda-määriteltävä termillä F. Oletetaan, että kohdassa $\boldsymbol{x} \in \mathbb{N}^n$ f saa arvon b. Lambda-määriteltävyyden mukaan

$$F\overline{\boldsymbol{x}} \overset{\beta}{=} \overline{b}, \quad \text{jos ja vain jos} \quad f(\boldsymbol{x}) = b.$$

Jos vakio l on termin F Gödel-luku, niin termin $F\overline{\boldsymbol{x}} = (\cdot\cdot((F\overline{x_1})\overline{x_2})\cdots\overline{x_n})$ Gödel-luku saadaan rekursiivisesti funktiolla

$$A(\boldsymbol{x}) = E\big(\cdots E\big(E(l * N(x_1)) * N(x_2)\big)\cdots N(x_n)\big).$$

Kaikilla $i \in \mathbb{N}$ Gödel-luku $\theta(A(\boldsymbol{x}), i)$ edustaa jotakin lambdatermin $F\overline{\boldsymbol{x}}$ kanssa beetayhtäsuurta termiä. Termin $\overline{b}$ Gödel-luku on $N(b)$. Lambda-määriteltävyyden nojalla on olemassa luku y, jolla $\theta(A(\boldsymbol{x}), y) = N(b)$. Churchin-Rosserin lauseen korollaarin (3.16) nojalla termi $F\overline{\boldsymbol{x}}$ ei sievene mihinkään toiseen (normaalimuotoiseen) kohdan 6 mukaiseen Churchin numeraaliin. Siten kyseinen y on ainoa luku, jolla relaatio

$$Num\big[\theta(A(\boldsymbol{x}), y)\big]$$

on voimassa. Tästä seuraa, että yhdistetty funktio

$$N^{-1}\big(\theta(A(\boldsymbol{x}), \mu y\big[Num[\theta(A(\boldsymbol{x}), y)]\big])\big) = N^{-1}(N(b)) = f(\boldsymbol{x})$$

on rekursiivinen. $\square$

Lauseiden (5.4) ja (5.5) myötä olemme todistaneet tutkielman peruslauseen.

Lause 5.1.

Funktio f on rekursiivinen, jos ja vain jos f on lambda-määriteltävä. $\square$

Kirjallisuutta

[1] Barendregt, Henk: Lambda Calculus, Its Syntax and Semantics (Studies in logic and the foundations of mathematics, volume 103), 1st edition, Elsevier Science Publishers B.V., 1989

[2] Church, Alonzo: An Unsolvable Problem of Elementary Number Theory, American Journal of Mathematics, Vol.58(2), 1 April 1936, s.345-363

[3] GeoGebra-web-sovelluksen WWW-sivu <https://www.geogebra.org/>. Luettu 30.10.2018

[4] Kleene, Stephen Cole: General recursive functions of natural numbers, Mathematische Annalen, Vol.112(1), 1936a, s.727-742

[5] Kleene, Stephen Cole: Lambda-definability and recursiveness, Duke Mathematical Journal 2, no. 2, 1936b, s.340-353

[6] Odifreddi, Piergiorgio: Classical Recursion Theory (Studies in logic and the foundations of mathematics, volume 125), 1st edition, Elsevier Science Publishers B.V., 1989

[7] Väänänen, Jouko: Matemaattinen logiikka, luentomoniste, Helsingin yliopisto, 2013